

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

DISERTACIÓN DE GRADO

DESARROLLO DE UN PROTOTIPO PARA
EL SEGUIMIENTO Y CUIDADO DE
JARDINES URBANOS BASADO EN LA
RECOPIACIÓN DE DATOS A TRAVÉS DE
LA PLATAFORMA DE ARDUINO

VALERIA ASTUDILLO
JAIME CADENA

AGRADECIMIENTOS

En forma muy especial al Ing. Andrés Jiménez quien con su conocimiento y experiencia supo guiarnos durante el desarrollo de esta disertación, además un agradecimiento al grupo de profesores que trabajaron en la revisión de este documento.

Jaime Cadena

AGRADECIMIENTOS

Agradezco a Dios por la vida que me regala y por la dicha de tener a mi lado a personas que hacen de mis días los mejores.

A mis padres por hacer de mí una persona íntegra, por confiar en mi esfuerzo y dedicación, por su paciencia y amor infinito. A mi familia por motivarme a seguir adelante y apoyarme siempre.

Expreso un profundo agradecimiento al Ing. Andrés Jiménez, quien nos brindó la oportunidad de hacer posible este trabajo de disertación, guiándonos y orientándonos con sus valiosos conocimientos. De igual manera, agradezco a los Ingenieros Fabián de la Cruz y Alfredo Calderón, por su aporte con la revisión y corrección.

Valeria Astudillo

DEDICATORIA

A mis padres por la formación
que son quienes han hecho
posible finalizar esta nueva meta

Jaime Cadena

DEDICATORIA

A mis padres Mario y Adriani,
quienes son el pilar de mi vida
y mi apoyo incondicional.

A mis hermanos Francisco y Paulina,
porque son mi motivación
para luchar por mis sueños.

A Carlos, por sus palabras
de aliento en cada momento.

Valeria Astudillo

Tabla de contenidos

CAPÍTULO 1: MARCO TEÓRICO	1
1.1: MODELOS PRESCRIPTIVOS DE PROCESO	1
1.1.1: INTRODUCCIÓN	1
1.1.2: DEFINICIÓN	2
1.1.3: MODELOS DE PROCESO INCREMENTAL.....	3
1.1.4: TIPOS DE MODELO INCREMENTAL	5
1.1.5: VENTAJAS DEL USO DEL MODELO INCREMENTAL.....	6
1.1.6: INCONVENIENTES DEL MODELO INCREMENTAL	7
1.2: TECNOLOGÍA ARDUINO	7
1.2.1: INTRODUCCIÓN	7
1.2.2: DEFINICIÓN ARDUINO.....	8
1.2.3: HISTORIA	8
1.2.4: LENGUAJE DE PROGRAMACIÓN PROCESSING/WIRING.....	10
1.2.5: VENTAJAS SOBRE OTRAS PLATAFORMAS	10
1.2.6: ARDUINO Y OTRAS TECNOLOGÍAS.....	12
1.3: ADMINISTRACIÓN DE JARDINES URBANOS.....	16
1.3.1: DEFINICIÓN JARDÍN URBANO	16
1.3.2: FUNCIONES.....	16
1.3.3: TIPOS DE JARDINES URBANOS	16
1.3.4: NOCIONES BÁSICAS DE UN JARDÍN URBANO	17
1.3.5: BUENAS PRÁCTICAS PARA EL CUIDADO	20
1.3.6: RESUMEN INDICADORES DE CUIDADO DE PLANTAS	21
1.3.7: PRINCIPALES INDICADORES DE DIFERENTES TIPOS DE PLANTAS.....	23
CAPÍTULO 2: ARQUITECTURA	27
2.1: DESCRIPCIÓN DEL PROBLEMA A SOLUCIONAR.....	27
2.2: FUNCIONALIDADES DEL PROTOTIPO	27
2.3: DIAGRAMA DE ARQUITECTURA GENERAL.....	28
2.3.1: DIAGRAMA DE ARQUITECTURA	28
2.3.2: UML DE PAQUETES.....	29
2.4: DESCRIPCIÓN DE COMPONENTES DE HARDWARE.....	29
2.4.1: SENSOR DE TEMPERATURA Y HUMEDAD DHT11	29
2.4.2: SENSOR DE MOVIMIENTO PIR HC-SR501	31
2.4.3: SENSOR DE LUZ LDR	33
2.4.4: SENSOR DE HUMEDAD DEL SUELO YL 38	34
2.4.5: PROTOBOARD	35
2.4.6: ARDUINO YÚN.....	35
2.5: DESCRIPCIÓN COMPONENTES DE SOFTWARE	39

2.5.1: ENTORNO ARDUINO	39
CAPÍTULO 3: PRIMERA ITERACIÓN ARDUINO Y SENSORES	42
3.1: ANÁLISIS	42
3.1.1: DESCRIPCIÓN DE REQUERIMIENTOS	42
3.1.2: CASOS DE USO.....	43
3.2: DISEÑO	50
3.2.1: DIAGRAMA DE ARQUITECTURA	51
3.3: CODIFICACIÓN	52
3.4: PRUEBAS.....	53
3.4.1: Sensor de humedad y temperatura	53
3.4.2: Sensor de luz.....	55
CAPÍTULO 4: SEGUNDA ITERACIÓN PÁGINA WEB	56
4.1: ANÁLISIS	56
4.1.1: DESCRIPCIÓN DE REQUERIMIENTOS	56
4.1.2: CASOS DE USO.....	57
4.2: DISEÑO	60
4.2.1: DIAGRAMA DE ARQUITECTURA	60
4.3: CODIFICACIÓN	61
4.3.1: Inicio.php.....	63
4.3.2: Recomendaciones.php	63
4.3.3: Tips.php.....	64
4.4: PRUEBAS.....	65
4.4.1: Diseño indicadores de cuidados de plantas	65
4.4.2: Diseño alertas de precauciones de cuidados de plantas	65
4.4.3: Diseño recomendaciones para sugerencia de plantas	67
4.4.4: Diseño tips caseros para cuidado de plantas	68
CAPÍTULO 5: INTEGRACIÓN PÁGINA WEB Y ARDUINO	69
5.1: ANÁLISIS	70
5.1.1: DESCRIPCIÓN DE REQUERIMIENTOS	70
5.1.2: CASOS DE USO.....	70
5.2: DISEÑO	73
5.2.1: DIAGRAMA DE ARQUITECTURA	74
5.3: CODIFICACIÓN	75
5.3.1: Codificación Arduino Yún	75
5.3.2: Codificación página web.....	75
5.4: PRUEBAS.....	80
5.4.1: Indicadores de cuidados de plantas	80
5.4.2: Alertas de precauciones de cuidados de plantas.....	81

5.4.3: Recomendaciones para sugerencia de plantas	83
5.4.4: Sensor de humedad y temperatura	84
5.4.5: Sensor de humedad del suelo.....	86
5.4.6: Sensor de Luz.....	86
CONCLUSIONES	88
RECOMENDACIONES	89
BIBLIOGRAFÍA	90
ANEXOS	95
ANEXO 1: CÓDIGO FUENTE SKETCH ARDUINO YÚN PRIMERA ITERACIÓN.....	95
ANEXO 2: CÓDIGO FUENTE PÁGINA WEB SEGUNDA ITERACIÓN.....	97
<i>Inicio.php</i>	97
<i>Recomendaciones.php</i>	99
<i>Tips.php</i>	103
ANEXO 3: CÓDIGO FUENTE SKETCH ARDUINO YÚN TERCERA ITERACIÓN	106
ANEXO 4: CÓDIGO FUENTE PÁGINA WEB TERCERA ITERACIÓN	108
<i>Inicio.php</i>	108
<i>Recomendaciones.php</i>	110
<i>RecomendacionFinal.php</i>	114
<i>Tips.php</i>	117
<i>Humedad.php</i>	119
<i>HumedadSuelo.php</i>	119
<i>Luz.php</i>	119
<i>Temperatura.php</i>	119
ANEXO 5: MANUAL DE USUARIO	120
<i>Requisitos de Software</i>	120
<i>Ingreso al sistema</i>	120
<i>Funcionalidades Generales</i>	121

Índice de figuras

Figura 1: Modelo de ciclo de vida en cascada.....	3
Figura 2: Modelo de ciclo de vida incremental	4
Figura 3: Arquitectura Java	13
Figura 4: Diagrama de arquitectura general	28
Figura 5: UML de paquetes general	29
Figura 6: Entorno de Arduino.....	40
Figura 7: Estructura de un programa en Arduino	41
Figura 8: Diagrama general primera iteración.....	43
Figura 9: Siguiendo nivel RF1.....	44
Figura 10: Siguiendo nivel RF2.....	44
Figura 11: Siguiendo nivel RF3.....	45
Figura 12: Diagrama consultar humedad ambiental.....	45
Figura 13: Diagrama consultar temperatura ambiental	47
Figura 14: Diagrama consultar luz ambiental	48
Figura 15: Diagrama consultar humedad suelo	49
Figura 16: Diagrama a detalle primera iteración	51
Figura 17: Diagrama general segunda iteración	57
Figura 18: Diagrama visualizar diseño de indicadores de cuidado de plantas	57
Figura 19: Diagrama visualizar diseño de alertas de precauciones de cuidado de plantas	58
Figura 20: Visualizar diseño de recomendaciones para sugerencia de plantas	59
Figura 21: Diagrama visualizar diseño de tips caseros para cuidado de plantas	60
Figura 22: Diagrama a detalle segunda iteración	61
Figura 23: Diagrama general tercera iteración	71
Figura 24: Diagrama visualizar indicadores de cuidado de plantas	71
Figura 25: Diagrama visualizar alertas de precauciones de cuidado de plantas.....	72
Figura 26: Visualizar recomendaciones para sugerencia de plantas	73
Figura 27: Diagrama a detalle tercera iteración.....	74

Índice de tablas

Tabla 1: Resumen indicadores de cuidado de plantas	22
Tabla 2: Tipos de plantas y sus indicadores	26
Tabla 3: Microcontrolador Arduino	37
Tabla 4: Microprocesador.....	37
Tabla 5: Especificaciones de tablas Arduino.....	37
Tabla 6: Primera prueba sensor de humedad y temperatura.....	54
Tabla 7: Segunda prueba sensor de humedad y temperatura.....	54
Tabla 8: Prueba sensor de luz	55
Tabla 9: Prueba diseño indicadores de cuidados de plantas	65
Tabla 10: Prueba diseño mensajes de alerta	66
Tabla 11: Prueba diseño alerta.....	67
Tabla 12: Prueba diseño recomendaciones para sugerencia de plantas.....	68
Tabla 13: Prueba diseño tips caseros para cuidado de plantas	69
Tabla 14: Prueba indicadores de cuidados de plantas	81
Tabla 15: Prueba mensajes de alerta.....	82
Tabla 16: Prueba alerta	83
Tabla 17: Prueba recomendaciones para sugerencia de plantas	84
Tabla 18: Tercera prueba sensor de temperatura.....	85
Tabla 19: Tercera prueba sensor de humedad	85
Tabla 20: Tercera prueba sensor de humedad del suelo.....	86
Tabla 21: Tercera prueba sensor de luz	87

CAPÍTULO 1: MARCO TEÓRICO

1.1: MODELOS PRESCRIPTIVOS DE PROCESO

1.1.1: INTRODUCCIÓN

En la actualidad, el software de computadora es la tecnología individual más importante en el ámbito mundial. También es uno de los ejemplos principales de la ley de las consecuencias imprevistas. Nadie en la década de los 1950 podría haber predicho que el software se convertiría en una tecnología indispensable en los negocios, la ciencia y la ingeniería; tampoco que el software permitiría la creación de tecnologías nuevas, el fin de tecnologías antiguas; que el software sería la fuerza conductora detrás de la revolución de las computadoras personales; que los productos empaquetados de software se podrían comprar en los centros comerciales; que una compañía de software se volvería muy grande y más influyente que la mayoría de las compañías de la era industrial; que una gran red construida con software llamada internet cubriría y cambiaría todo, desde la investigación bibliográfica hasta las compras de los consumidores y los hábitos diarios de los jóvenes y no tan jóvenes.

Nadie podría haber previsto que el software estaría relacionado con sistemas de todo tipo: de transporte, médicos, de telecomunicaciones, militares, industriales, de entrenamiento, máquinas para oficina.

Por último, nadie podría haber predicho que millones de programas de computadora tendrían que corregirse, adaptarse y mejorarse conforme pasara el tiempo y que la labor de desarrollar estas actividades de mantenimiento absorbería más gente y recursos que todo el trabajo aplicado para la creación del software nuevo. (Roger, 2005)

Modelos y metodologías de desarrollo fueron propuestas para poder tener una administración y control del desarrollo, modificación o actualización de un producto o

sistema. Además de permitir a los equipos de trabajo tener indicadores de medidas del desempeño, estado y calidad del producto o sistema que se encuentra en desarrollo.

1.1.2: DEFINICIÓN

Los modelos prescriptivos de proceso se propusieron originalmente para ordenar el caos del desarrollo de software. La historia ha indicado que estos modelos convencionales han traído consigo cierta cantidad de estructuras útiles para el trabajo en la ingeniería del software, y han proporcionado un camino a seguir razonablemente efectivo para los equipos de software. Sin embargo, el trabajo de la ingeniería del software y el producto resultante aún permanecen al borde del caos. (Roger, 2005)

Este tipo de modelos proporcionaron al desarrollo de productos y sistemas un control y organización que permitió que los ingenieros de software puedan satisfacer de una mejor manera las necesidades y requerimientos de las personas del negocio, cumpliendo con una alta calidad en el producto final.

Los procesos prescriptivos de procesos son guías y referencia para todas las etapas del desarrollo de software. “No proveen una respuesta definitiva a los problemas de desarrollo de software en un entorno que cambia permanentemente” (Perez, 2011). Por lo que los procesos para el desarrollo de software tienen que adaptarse a cada caso individual, en donde se tienen equipos y ambientes de trabajo distintos.

Dentro de los modelos prescriptivos de proceso se pueden categorizar dos tipos dependiendo de cómo basan su flujo de trabajo durante el desarrollo del producto o servicio.

1.1.2.1: Modelo en Cascada

Se cree que el modelo en cascada fue el primer modelo de proceso introducido y seguido ampliamente en la ingeniería el software. La innovación estuvo en la primera vez que la ingeniería del software fue dividida en fases separadas.

La primera descripción formal del modelo en cascada se cree que fue en un artículo publicado en 1970 por Winston W. Royce. (Ingeniería del software: Metodologías y ciclos de vida, 2009)

El modelo en cascada se caracteriza porque avanza de una etapa a otra de manera secuencial, como se puede apreciar en la Figura 1. Por lo que se aconseja que se utilice para proyectos de desarrollo de software en los que los requerimientos de los usuarios están claramente definidos y no serán modificados durante etapas posteriores.

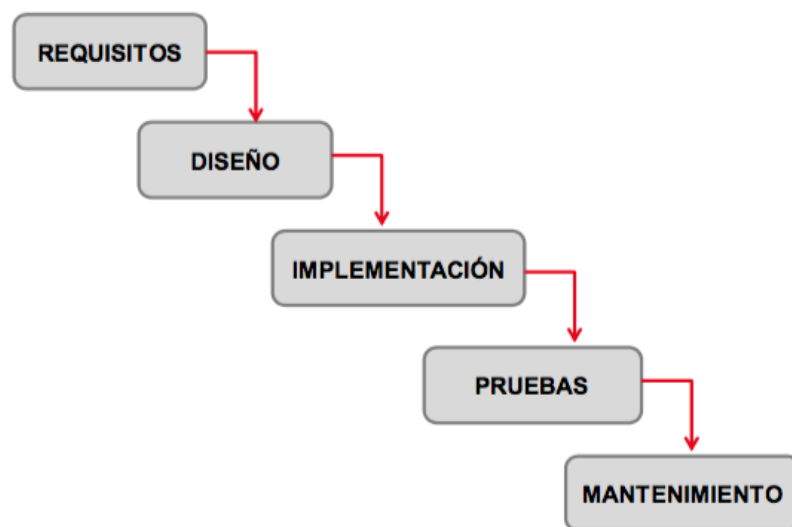


Figura 1: Modelo de ciclo de vida en cascada
Fuente: (Ingeniería del software: Metodologías y ciclos de vida, 2009)

1.1.2.2: Modelo de proceso incremental

En muchas situaciones los requisitos iniciales del software están bien definidos en forma razonable, pero el enfoque global del esfuerzo de desarrollo excluye un proceso puramente lineal. Además, quizá haya una necesidad imperiosa de proporcionar de manera rápida un conjunto limitado de funcionalidad para el usuario, después refinarla y expandirla en las entregas posteriores de software. En estos casos se elige un modelo de proceso diseñado para producir el software en forma incremental. (Roger, 2005)

1.1.3: MODELOS DE PROCESO INCREMENTAL

El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. (Ingeniería del software: Metodologías y ciclos de vida, 2009). Como se puede ver la Figura 2.

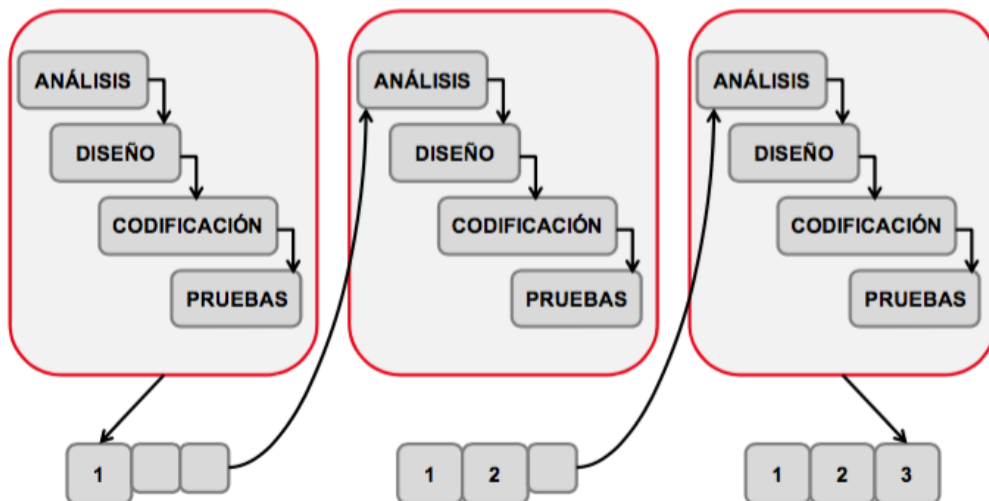


Figura 2: Modelo de ciclo de vida incremental
Fuente: (Ingeniería del software: Metodologías y ciclos de vida, 2009)

El modelo incremental está enfocado principalmente a casos que, durante el desarrollo de software, es necesario presentar un entregable al usuario en un plazo de tiempo corto, en este entregable no se considera que se disponga de funcionalidades limitadas. El objetivo de realizar el desarrollo de software con este enfoque, es el de poder satisfacer necesidades imperiosas del usuario. El software entregado inicialmente al usuario será refinado y se le añadirán funcionalidades en las posteriores versiones o entregas.

Cuando se utiliza el modelo incremental para el desarrollo de un nuevo software, el primer incremento que se desarrolla es generalmente un producto básico, contiene los requerimientos principales recolectados del usuario.

El modelo incremental se centra en la entrega de un producto operativo con cada incremento presentado. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación de la calidad.

1.1.4: TIPOS DE MODELO INCREMENTAL

1.1.4.1: Modelo Incremental

“El modelo incremental combina elementos del modelo en cascada aplicado en forma iterativa” (Roger, 2005). Una de las principales características del modelo incremental es entregar al cliente un producto al cual evaluar. Las siguientes iteraciones a realizarse tiene en consideración el feedback presentado por el usuario del anterior producto evaluado, con el siguiente entregable se busca satisfacer de mejor manera los requerimientos del cliente, además de incluir funcionalidades adicionales.

Se debe considerar que los primeros entregables de las primeras iteraciones son versiones incompletas del producto o servicio final, que están enfocadas a entregar la funcionalidad requerida al usuario. “El Modelo Incremental es de naturaleza iterativa brindando al final de cada incremento la entrega de un producto completamente operacional” (Procesossoftware, n.d.).

1.1.4.2: Modelo DRA

El desarrollo rápido de aplicaciones (DRA) es un modelo de proceso de software incremental que resalta un ciclo de desarrollo corto. El modelo DRA es una adaptación a la alta velocidad del modelo en cascada en el que se logra el desarrollo rápido mediante un enfoque de construcción basada en componentes. (Roger, 2005)

El modelo DRA está principalmente enfocado para que un equipo de desarrollo sea capaz de realizar un sistema completamente funcional dentro de un período de tiempo relativamente corto.

El desarrollo rápido de aplicaciones está basado en fases claramente definidas para cada proceso necesario a realizarse para el desarrollo.

La comunicación trabaja para entender el problema de negocios y las características de información que debe incluir el software. La planeación es esencial porque varios equipos de software trabajan en paralelo sobre diferentes funciones del sistema. El modelado incluye tres grandes fases – modelado de negocios, modelado de datos y modelado del proceso – y establece representaciones del diseño que sirven como base para la actividad de construcción del modelo DRA. La construcción resalta el empleo de componentes de software existente y la aplicación de la generación automática de código. Por último, el despliegue establece una base para las iteraciones subsecuentes, si éstas son necesarias. (Roger, 2005)

Los principales inconvenientes del enfoque DRA son:

- Para proyectos grandes, pero escalables, el DRA necesita suficientes recursos humanos para crear el número correcto de equipos DRA.
- Si los desarrolladores y clientes no se comprometen con las actividades rápidas necesarias para completar el sistema en un marco de tiempo muy breve, los proyectos de DRA fallarán.
- Si un sistema no se puede modelar en forma apropiada, la construcción de los componentes necesarios para el DRA será problemática.

1.1.5: VENTAJAS DEL USO DEL MODELO INCREMENTAL

Entre las ventajas que puede proporcionar el utilizar el modelo incremental para el desarrollo de nuevo software tenemos:

- Se puede generar software operativo de forma rápida y en etapas tempranas del ciclo de vida del software.
- Es un modelo flexible, por lo que se puede realizar cambios en el alcance y requisitos.
- Realizar pruebas y depuración son más sencillas al hacerse en pequeñas iteraciones.
- Facilidad para gestionar riesgos.

1.1.6: INCONVENIENTES DEL MODELO INCREMENTAL

Uno de los principales limitantes que se tiene para utilizar este modelo para el desarrollo de un nuevo software es que requiere experiencia para definir los incrementos y distribuir en ellos las tareas de forma proporcionada.

Otros de los posibles inconvenientes a presentarse son:

- Las fases de una iteración son rígidas e interactúan con otras.
- Problemas relacionados con la arquitectura del sistema, porque no todos son definidos durante la primera iteración.

1.2: TECNOLOGÍA ARDUINO

1.2.1: INTRODUCCIÓN

Actualmente en el mercado tecnológico, existen varias plataformas de hardware libre que se han desarrollado con el propósito de brindar a sus usuarios la facilidad de estudiarlas, modificarlas, de crear nuevas plataformas a partir de sus diseños disponibles, sin costo alguno y con un total acceso.

Gracias a estas plataformas, se ha impulsado tanto a profesionales como a estudiantes, a la creación de proyectos innovadores bajo la idea de “hágalo usted mismo”, o mejor conocida como “Do It Yourself”. De esta manera se busca promover nuevas propuestas de proyectos que se desarrollen a partir del ingenio de los usuarios, con el fin de que éstas sean compartidas y puedan estar al alcance de los demás para mejorarlas cuando sea adecuado.

Cada plataforma de hardware libre cuenta con diferentes componentes y especificaciones propias, unas más avanzadas que otras, pero todas brindando facilidad de aprendizaje y manejo para los distintos propósitos de uso.

1.2.2: DEFINICIÓN ARDUINO

Al momento de mencionar a Arduino, se puede pensar en una placa de circuitos electrónicos que da la oportunidad de poner en marcha cualquier tipo de proyecto. En sí, Arduino es una plataforma electrónica de código abierto que se fundamenta en un trabajo en conjunto tanto de software como de hardware, con el fin de crear un todo acorde a las necesidades de un usuario final. “La acogida que ha tenido Arduino ha generado que muchas de esas empresas que jamás se cuestionaron la posibilidad de hacer objetos libres, estén haciéndolos libres” (Sánchez Onofre, 2014).

“Arduino es una plataforma de hardware y software de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing” (Amangandi, 2012).

1.2.3: HISTORIA

El surgimiento de esta tecnología se da en el año 2005, gracias a la necesidad de un estudiante del instituto IVREA de adquirir mejores y mayores conocimientos en cuanto a la electrónica y la computación. En aquella época, poder contar con una herramienta que contenga microcontroladores y circuitos eléctricos era realmente difícil, más que nada por el costo elevado. Es por eso que Arduino nace como una idea de proyecto escolar, para luego convertirse en una herramienta líder en el ámbito tecnológico.

Los componentes del primer prototipo de Arduino fueron una placa pre ensamblada de circuitos eléctricos junto con un microcontrolador compuesto de resistencias. Como característica principal es que tenía la capacidad de conectarse con sensores de categoría simple, como por ejemplo pequeños leds.

Este proyecto con el pasar de los años fue llamando la atención de varios estudiantes que se convertirían en colaboradores vitales para el desarrollo y avance de esta tecnología. El equipo denominado “Team Arduino” fue conformado por estudiantes de institutos italianos, colombianos, españoles y estadounidenses.

El primer instituto aportó tanto con el desarrollo del ambiente para la programación del procesador de la placa, como con la mejora de la interfaz del software. El segundo instituto propone un progreso en cuanto al hardware de la placa, al agregar microcontroladores que ofrezcan una memoria superior para el ambiente de programación. El tercer instituto mejoró Arduino al potenciarlo con puertos USB con el fin de poder realizar conexiones directas a ordenadores y así llegar a la idea de una distribución de dicho proyecto.

Teniendo en cuenta que la placa final contaba con lo necesario para ser una tecnología eficaz, se comienza con su distribución, en primera instancia de manera gratuita en el instituto IVRAE en donde tuvo una notoria acogida tanto en estudiantes como en profesores. Poco tiempo después se proyectó en una distribución a nivel mundial, comenzando en países como Italia y España, donde se percibió una gran aceptación, a tal punto de convertirse en una de las mejores herramientas de aprendizaje, con un total acceso y a un precio conveniente. (Arduino: Tecnología para todos, s.f.). Díaz (2014) afirma que:

Massimo Banzi ocupa, con el resto de su equipo, una posición muy destacada al ser uno de los padres del proyecto Arduino que desde unos inicios muy modestos ha hecho que el Open Hardware deje de ser un concepto de nicho para estar al alcance de todo el mundo.

Así, el “Team Arduino”, quienes tuvieron la posta de la creación y desarrollo de este gran proyecto, está conformado por:

- Massimo Banzi (Italia)
- Hernando Barragán (Colombia)
- David Mellis (Estados Unidos)

- David Cuartielles (España)
- Tom Igoe (Estados Unidos)
- Gianluca Martino (Italia)

1.2.4: LENGUAJE DE PROGRAMACIÓN PROCESSING/WIRING

El lenguaje con el que trabaja la plataforma Arduino, es el lenguaje de programación Arduino que se basa en Wiring y el entorno de desarrollo Arduino que se basa en Processing.

Processing no es solamente un lenguaje de programación orientado a la creación visual, si no toda una plataforma de creación de aplicaciones computacionales altamente interactivas que no solo se alimentan de los eventos producidos por los usuarios, si no por una gran cantidad de librerías que permiten conectarse a un sin número de plataformas y aplicativos que enriquecen las aplicaciones y le permiten producir una gran cantidad de acciones sobre el mundo virtual de la computadora. (Revuelta, 2011)

En cuanto Wiring, se centra en ofrecer un marco de programación especialmente para microcontroladores, proyecto abierto que fue iniciado por Hernando Barragán en el instituto IVREA con el ideal de que estudiantes, diseñadores y expertos fomenten su aprendizaje y se motiven en la creación de nuevas ideas.

“Wiring es un entorno de programación de entradas/salidas de código abierto para explorar las artes electrónicas, los medios materiales, la enseñanza y el aprendizaje de la programación informática y creación de prototipos con electrónica” (Revuelta, 2011).

1.2.5: VENTAJAS SOBRE OTRAS PLATAFORMAS

La plataforma Arduino, además de prestar la facilidad de realizar un sin número de proyectos de distinto rango, como por ejemplo: “desde robótica hasta domótica, pasando por monitorización de sensores ambientales, sistemas de navegación,

telemática, etc.” (Torrente Artero, 2013, pág. 66), ofrece ciertas ventajas que son un factor diferencial frente a otras plataformas:

- **Costo accesible:** en comparación con otras plataformas, se puede decir que comprar una placa Arduino es realmente barato, dependiendo siempre y cuando de sus componentes. “La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50 dólares” (Enríquez Herrador, 2009, pág. 9).
- **Multiplataforma:** mientras otros microcontroladores tan sólo se ejecutan en Windows, Arduino tiene la capacidad de ejecutarse en sistemas operativos como: Windows, GNU/Linux y Macintosh OSX.
- **Entorno de programación flexible:** “El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también” (Enríquez Herrador, 2009, pág. 9). Al ser Processing un lenguaje de programación similar a C++, los estudiantes pueden familiarizarse de mejor manera y escribir con facilidad líneas de código en este entorno.
- **Código abierto:** Arduino es una herramienta de código abierto que da la posibilidad a que estudiantes, profesores o simplemente personas experimentadas con esta plataforma, puedan añadir, modificar, crear, mejorar funcionalidades de proyectos junto a este entorno.
- **Hardware y software extensible:** “Cualquiera que desee ampliar y mejorar tanto el diseño hardware de las placas como el entorno de desarrollo software y el propio lenguaje de programación puede hacerlo sin problemas” (Torrente Artero, 2013, pág. 70). El software puede expandirse por medio de librerías disponibles en C++, mientras que el hardware puede ser expandido a través de nuevas versiones de las placas construidas ya sea por diseñadores o inexpertos.

1.2.6: ARDUINO Y OTRAS TECNOLOGÍAS

1.2.6.1: INTRODUCCIÓN

De acuerdo a cuáles sean las necesidades del problema o proyecto que se desea resolver, se puede aprovechar la gran compatibilidad de comunicación que ofrece Arduino. Por ello existe la posibilidad de interactuar con diferentes sistemas y lenguajes.

Es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, como por ejemplo Java, Flash, Pure Data, etc., gracias a que se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes que se mencionaron anteriormente soportan.

Además, gracias a su estructura GNU/Linux, se pueden montar servidores como HTTP, FTP o SSH entre otros, y utilizar PHP o Python paralelo al uso de las ventajas que ofrece el microcontrolador incorporado.

1.2.6.2: JAVA

Creado por Sun Microsystems Inc. en 1990, como un proyecto para desarrollar un sistema que controlara electrodomésticos. Se pretendía crear un hardware polivalente, con un Sistema Operativo eficiente (SunOS) y un lenguaje de desarrollo denominado Oak (roble), el proyecto finalizó en 1992 y debido a su alto costo en relación a otros proyectos del mismo tipo que ya existían, fue declarado como un fracaso. (Lopez, n.d.)

Java es un lenguaje de programación cuya principal característica es de ser un lenguaje compilado e interpretado, es decir, que todo software desarrollado utilizando este lenguaje el código debe compilarse, se debe generar bytecodes para que pueda ser interpretado por una máquina virtual, que posteriormente va a ser interpretado por la computadora.

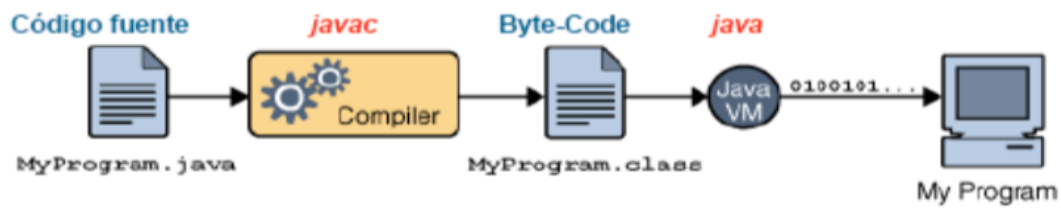


Figura 3: Arquitectura Java
Fuente: (Lopez, n.d.)

Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto. (Lopez, n.d.)

Java, al estar construido bajo esta arquitectura permite que el software desarrollado tenga independencia al sistema operativo en el cual se está ejecutando porque el código está compilado y es ejecutado en máquinas virtuales que están disponibles en las principales plataformas.

El nombre del lenguaje tuvo que ser cambiado ya que existía otro lenguaje llamado Oak. Se buscaban un nombre que evocara la esencia de la tecnología (viveza, animación, rapidez, interactividad...). Java fue elegido de entre muchísimas propuestas. No es un acrónimo, sino más bien un nombre inspirado en algo que a muchos programadores les gusta beber en grandes cantidades: una taza de café (Java en argot Inglés americano). De esta forma, Sun lanzó las primeras versiones de Java a principios de 1995. (Lopez, n.d.) (Historia PHP, n.d.)

Java puede comunicarse con la placa Arduino con el uso del puerto serial, para poder permitir que Java se comunique con este puerto es necesario que se utilice una librería externa, llamada RXTX Java Library.

Ardulink es una alternativa para poder interactuar con la placa Arduino desde Java, esta solución define protocolos para poder realizar la comunicación

“Ardulink es una solución completa, open source, es una solución para java que permite el control y coordinación de placas Arduinos. Esta solución define un protocolo de

comunicación y una interfaz de comunicación que permite la implementación de algunos protocolos” (Arduino, n.d.).

1.2.6.3: PYTHON

Es un lenguaje de programación de alto nivel y de propósito general que se caracteriza por permitir múltiples paradigmas de programación. Python es gestionado por la organización Python Software Foundation.

Es un lenguaje de muy alto nivel que permite expresar algoritmos de forma casi directa y hemos comprobado que se trata de un lenguaje particularmente adecuado para la enseñanza de la programación. Esta impresión se ve corroborada por la adopción de Python como lenguaje introductorio en universidades.

Python ha sido diseñado por Guido van Rossum y está en un proceso de continuo desarrollo por una gran comunidad de desarrolladores. Aproximadamente cada seis meses se hace pública una nueva versión de Python. No es que cada medio año se cambie radicalmente el lenguaje de programación, sino que este se enriquece manteniendo en lo posible la compatibilidad con los programas escritos para versiones anteriores. (Historia PHP, n.d.)

Python tiene diversas herramientas y librerías que le puede ayudar a leer los datos que son enviados desde la placa Arduino. La librería pySerial no solamente está enfocada a recibir los datos enviados por placas Arduino, sino que por otros dispositivos que utilicen un puerto serial para el envío de datos.

“Este módulo contiene los acceso para el puerto serial, provee un backend para Python en Windows, OSX, Linux, BSD y IronPython” (Liechti, n.d.).

Además de librerías para aprovechar las placas Arduino, existen API's para poder desarrollar una aplicación de Arduino basada en Python, como es el caso de Python Arduino Prototyping.

1.2.6.4: PHP

“PHP es un lenguaje para el desarrollo de aplicaciones en la Web” (Cabezas Granado & González Lozano, 2015, pág. 19). Welling & Thomson (2005) afirman que:

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutará cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.

(...) Las siglas PHP equivalían inicialmente a Personal Home Page (Página de inicio) pero se modificaron de acuerdo con la convención de designación de GNU (del inglés, Gnu's Not Unix, Gnu no es Unix) y ahora equivale a PHP Hipertext Preprocessor (Preprocesador de hipertexto PHP). (pág. 33)

La manera más sencilla para poder interactuar desde PHP con placas Arduinos, es utilizando la librería PHP Serial Class, con la cual se puede realizar de una manera sencilla la comunicación con placas Arduino mediante los puertos seriales.

1.2.6.5 MySQL

MySQL es una base de datos de código abierto, es decir, cualquier persona está en la capacidad para poder contribuir a MySQL, resolviendo problemas, aumentando funcionalidades u optimizaciones, todo esto es porque el código fuente de MySQL está disponible.

“Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos” (Gilfillan).

El método más común para almacenar datos que se obtengan de una placa Arduino a una base de datos MySQL es utilizando otro lenguaje de programación para obtener los datos y enviarlos a la base de datos.

La librería MySQL Connector/Arduino es una alternativa que permite la comunicación con MySQL sin la necesidad de utilizar terceros para poder obtener los datos.

“La librería Connector/Arduino permite ejecutar queries a una base de datos, de una manera similar a la que sería mediante el cliente MySQL. Se puede insertar, eliminar,

actualizar y actualizar los datos, llamar funciones, crear objetos, etc” (Introducing MySQL Connector/Arduino, 2013).

1.3: ADMINISTRACIÓN DE JARDINES URBANOS

1.3.1: DEFINICIÓN JARDÍN URBANO

Un jardín urbano es “un espacio cubiertos o no destinados para el cultivo de flores, plantas aromáticas, hortalizas, hierbas medicinales frutales, a escala doméstica” (Huerto urbano en mi balcón, 2015). Los jardines urbanos hoy en día son considerados como parte del equilibrio ambiental y social.

1.3.2: FUNCIONES

Según López Medina (2008):

Los parques y jardines urbanos constituyen, (...) el único reducto que los ciudadanos tienen para aliviarse de los males endémicos de las ciudades. Son una garantía de la presencia de la naturaleza en la ciudad, cumplen funciones muy diversas, unas de índole práctico y otras de índole psicológica. Contribuyen con funciones ornamentales, proporcionan espacios recreativos y de expansión, mejoran las condiciones micro climáticas de la ciudad, contribuyen a reducir la contaminación ambiental e influyen de forma positiva, tanto en la salud física como en la salud mental de los seres humanos que viven en un entorno cada vez más urbanizado.

1.3.3: TIPOS DE JARDINES URBANOS

Cada jardín urbano tiene un peculiar aspecto que los diferencia unos de otros. Los distintos tipos de jardines urbanos se los define de acuerdo a su ubicación o localización:

- Jardín urbano ubicado en la ciudad pero alejado del campo.

- Jardín urbano ubicado en los patios o jardines de las casas.
- Jardín urbano ubicado en los balcones y terrazas de los edificios.

1.3.4: NOCIONES BÁSICAS DE UN JARDÍN URBANO

Los principios que corresponden a la creación y mantenimiento de un jardín urbano, se deben generalmente a factores externos, los cuales se ponen a consideración.

1.3.4.1: Luz y sombra

En un jardín urbano es de vital importancia encontrar un equilibrio entre la luz y sombra, teniendo en cuenta el tipo de plantas que se tiene a cargo, ya que mientras unas necesitan de abundante luz solar, otras se mantienen con una menor cantidad. “En un jardín el contraste entre luz resplandeciente y sombra puede tener muchas cualidades distintas” (Wilde, 2008, pág. 16). Boisset (1995) afirma que:

La luz solar es esencial, ya que la clorofila en las células de las plantas la absorbe para convertir el agua absorbida por las raíces y el dióxido de carbono absorbido por las hojas en azúcar y agua. (...) Por lo general se podría afirmar que cuanto más soleado sea el lugar donde se halla la planta, mayor será su potencial; pero hay muchas excepciones. (pág. 16)

Existen distintos tipos de sombra que se deben considerar al momento de mantener un jardín urbano:

- **Sombra parcial:** es aquella que varía debido a un muro o edificio, es decir, la sombra puede perpetuarse durante unas horas en el día en una determinada zona del jardín, y otras horas dicha zona ya queda expuesta a la luz solar.
- **Sombra plena:** es aquella que brinda una sombra completa a diferentes zonas del jardín, a pesar de que reciben una luz solar difuminada que es suficiente para que las plantas crezcan de manera óptima.

- **Sombra densa:** es aquella que resulta excesiva, en mucho de los casos, causante de grandes problemas. Aun así existen plantas que pueden adaptarse a este tipo de sombra, aunque no crezcan de la misma forma que deberían crecer si éstas recibieran la luz solar necesaria (Boisset, 1995, pág. 16).

1.3.4.2: Clima y tiempo

El clima se lo puede considerar como uno de los factores más influyentes en el cuidado y mantenimiento de un jardín urbano, pues se convierte en una de las pautas para saber qué tipo de plantas cultivar y determinar cuáles de ellas necesitan más cuidados que otras. Existen condiciones climáticas que se deben tener en cuenta como: la temperatura, lluvia y humedad (Wilde, 2008, pág. 22).

1.3.4.3: Temperatura

La temperatura es el factor de mayor importancia pues representa una influencia directa con el proceso de crecimiento de las plantas. Un jardín urbano se puede exponer tanto a la temperatura del aire como la del suelo.

- **Temperatura del aire:** es un factor que influye con el desarrollo de las plantas que son parte de un jardín urbano. “La temperatura del aire cobra importancia cuando comienza el nuevo crecimiento de brotes; aquí, las temperaturas más tibias estimulan el crecimiento más rápido de la parte superior” (Buechel, 2015). Además, “la temperatura óptima para una planta depende de varios factores, y uno de ellos es el estado de desarrollo de la planta, ya que las plantas tienen una especie de reloj biológico que determina su sensibilidad a la temperatura” (CANNA, 2015).
- **Temperatura del suelo:** según Almorox (2007) “el calentamiento del suelo dependerá de la cantidad de radiación neta que llegue a la superficie terrestre resultado de considerar el balance energético de onda corta y de onda larga”.

Al igual que la temperatura del aire, la temperatura del suelo varía de acuerdo a cambios diarios o estacionales, por lo que ambas están relacionadas. “La temperatura del suelo puede ser un factor limitante para la germinación de la semilla, crecimiento de las raíces, desarrollo de los tubérculos, descomposición de la materia orgánica dentro del suelo y por lo tanto para la cantidad de dióxido de carbono que pasa del suelo a la capa planta-aire y de ésta a la atmósfera” (Ibáñez, 2008).

1.3.4.4: Lluvia

La generación de lluvia, depende de la región en la que el jardín urbano se encuentre. Por lo general los jardines reciben el agua necesaria de las lluvias que ocurren regularmente, pero a ciencia cierta no se conoce en qué intervalo de tiempo ocurrirán, ni qué cantidad de lluvia caerá. Según Wilde (2008), esto puede representar dos problemas principales:

- El encharcamiento del suelo, que tiene lugar cuando una fuerte lluvia prolongada cae sobre un terreno de drenaje pobre.
- Las sequías, allí donde la lluvia cae de forma muy escasa durante largos períodos ocasionando que el suelo se seque y endurezca. (pág. 24)

1.3.4.5: Humedad

La humedad es la cantidad de vapor de agua que hay en la atmósfera. Ésta se ve afectada también por el contenido de humedad en el suelo. Una humedad elevada puede fomentar el crecimiento de moho y enfermedades ocasionadas por hongos, mientras que una humedad escasa puede hacer que las plantas se sequen y marchiten con mayor rapidez. (Wilde, 2008, pág. 25)

Al igual que en la temperatura, es necesario tomar en cuenta tanto la humedad del aire como la del suelo:

- **Humedad del aire:** es afectada por el medio ambiente que rodea el jardín urbano. “Las condiciones de alta humedad son más prevalentes en la noche en climas fríos o también durante el día en estaciones frías. En la noche la humedad relativa puede hacerse muy alta con el descenso de la temperatura en el anochecer y mantenerse alta hasta la mañana cuando el calor del sol caliente” (Giraldo, 2011).
- **Humedad del suelo:** la humedad del suelo es un concepto esencial ya sea en cultivos intensivos, extensivos, huertos ecológicos, las plantas de una casa y todo aquello que tenga que ver con desarrollo vegetal. (...) Es sencillo pensar que la humedad del suelo va a ser simplemente el agua que contenga el suelo (Agromática, 2014).

Reyna, y otros, (2011) afirman que:

La determinación de la humedad natural del suelo es primordial para resolver problemas vinculados a las necesidades de agua de riego, consideraciones ambientales y determinación de los excedentes hídricos. (pág.91)

1.3.5: BUENAS PRÁCTICAS PARA EL CUIDADO

Para lograr un mantenimiento óptimo y correcto de un jardín urbano, se deben seguir ciertos cuidados para todos los ámbitos que lo rodea. No es cuestión de que el cuidado se realice de una forma momentánea, sino que sea continua y sobretodo sea el adecuado.

1.3.5.1: Luz

- La mayoría de plantas necesitan estar expuestas al sol por lo menos seis horas. Cuando se trata de plantas que son jóvenes o necesitan de un cuidado especial, es recomendable ubicarlas en una sombra temporal.
- El saber diferenciar y conocer qué plantas necesitan menor o mayor exposición solar, es vital.

1.3.5.2: Temperatura

- Escoger las plantas que serán parte del jardín urbano dependiendo de la región en la que se encuentre.
- Si el clima es frío, reubicar las plantas junto a un muro ya que éste en las horas del día acumula el calor y en la noche lo proporciona.
- Tener un control de la temperatura del suelo es importante, ya que si éste está debajo de los siete grados centígrados, las raíces pueden tener un crecimiento lento y será un limitante para la absorción de agua y nutrientes (Infojardín, 2015).

1.3.5.3: Riego

- El riego está relacionado con la humedad del suelo, por lo que se debe evitar el exceso de agua para que las plantas no se pudran.
- El riego se lo debe realizar en toda la planta, no sólo en una parte, para que pueda absorber por completo y penetre en todo el suelo. (Cantudo, 2011)
- “Para evitar quemaduras en las hojas y ayudar a que la tierra conserve más tiempo la humedad, riega a primera hora de la mañana o mejor al atardecer, pero nunca en las horas centrales del día” (Ecoterrazas, 2013).
- “La tierra absorbe mejor el agua y mantiene las raíces suficientemente húmedas si el riego es lento, como se puede lograr con mangueras de goteo o con las diseñadas para empapar una zona” (Grajales-Hall, 2002).

1.3.6: RESUMEN INDICADORES DE CUIDADO DE PLANTAS

Los principales indicadores de cuidado de plantas a tomar en cuenta, se resumen en la Tabla 1. Dichos indicadores constituyen una pauta para la definición de las variables que se deberán utilizar para la implementación del prototipo.

Indicador	Descripción
Humedad ambiental	Indicador que depende del medio ambiente. A través de la humedad ambiental las plantas tienden a transpirar y hacer uso del vapor de agua generada por la misma. Es esencial el control de humedad ambiental tanto excesiva, como escasa.
Temperatura ambiental	Indicador que afecta directamente el crecimiento de las plantas. Existen plantas que soportan temperaturas mayores a otras. La temperatura ambiental puede influir en cualquier etapa de desarrollo de la planta.
Luz ambiental	Indicador que influye en el florecimiento de las plantas. Aunque existen excepciones de plantas que no son aptas para recibir abundante luz, la mayoría de ellas la aprovechan al máximo.
Humedad del suelo	Indicador considerado como uno de los más importantes para el desarrollo de las plantas. Controlar la humedad del suelo es esencial para la transpiración de las plantas.
Movimiento	Indicador que permite la detección de animales u otros elementos que se encuentren en el medio ambiente, los mismos que se pueden considerar como una amenaza para el desarrollo óptimo de las plantas.

*Tabla 1: Resumen indicadores de cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)*

1.3.7: PRINCIPALES INDICADORES DE DIFERENTES TIPOS DE PLANTAS

Un jardín urbano está compuesto por varios tipos de plantas, ya sean medicinales, aromáticas, entre otras. A continuación se mencionarán las plantas consideradas como las más comunes dentro de un jardín urbano, incluyendo los indicadores de humedad y temperatura ambiental, luz solar, humedad del suelo, a los que deben estar expuestos para su mantenimiento:

Nombre planta	Luz	Humedad	Riego	Temperatura
Violeta	Necesita recibir abundante luz.	Es necesario una atmósfera húmeda.	Regar 2 veces a la semana en verano, y en invierno una vez cada 15 días.	Temperatura mínima de invierno a 12°C.
Rosa	Necesita recibir abundante luz.	Es necesaria una atmósfera húmeda en el verano.	Regar cada 10 o 15 días durante el crecimiento y en invierno mantener seco.	Temperatura entre 12°C y 15°C en una situación luminosa.
Geranio	Necesita luz, pero en horas centrales del día proteger del sol.	Necesaria una atmósfera húmeda moderada.	En el verano regar 2 veces por semana y en invierno apenas regarla.	No resiste a una temperatura por debajo de -2°C.
Helecho	Necesita recibir abundante luz.	Humedad mayor al 50%	En el verano regar 2 veces por semana y en invierno 1	Temperatura superior a los 7°C.

			vez por semana.	
Anturio	Necesita recibir abundante luz.	Humedad elevada.	En verano regar 3 veces a la semana y en invierno 1 vez por semana.	Temperatura mínima 15°C y máxima 29°C.
Dalia	Necesita recibir abundante luz.	Humedad moderada.	Regar 3 veces por semana	Temperatura entre 16°C y 20°C.
Pensamiento	Necesita recibir abundante luz.	Humedad moderada.	En verano regar 2 veces por semana.	Temperatura entre 16°C y 18°C.
Clavel	Necesita recibir abundante luz.	Humedad elevada.	En verano regar 3 veces a la semana.	Temperatura en el día entre 22°C y 24°C, en la noche entre 10°C y 12°C.
Hortensia	Necesita recibir poca luz.	Humedad moderada.	Regar 2 veces por semana.	Temperatura no menos de -3°C.
Margarita	Al menos 5 horas de luz solar.	Humedad elevada.	En verano regar 2 veces por semana y en invierno 1 vez por semana.	Temperatura no menos de -2°C.
Gladiolo	Necesita recibir abundante luz.	Humedad moderada.	Regar 3 veces por semana.	Temperatura en el día entre 20°C y 25°C, en la noche entre 10°C y

				15°C
Manzanilla	Necesita recibir abundante luz.	Humedad moderada.	Regar 3 veces por semana.	Temperatura mínima 13°C y máxima 20°C.
Cartucho	Necesita recibir poca luz.	Humedad moderada.	Regar 3 veces por semana.	Temperatura mínima 10°C y máxima 16°C.
Hiedra	Necesita recibir una luz moderada.	Humedad moderada.	En verano regar 2 o 3 veces por semana y en invierno cada 7 o 10 días.	Temperatura mínima de 8°C.
Begonia	Necesita recibir una luz moderada.	Necesita una alta humedad.	En verano regar 2 veces por semana y en invierno cada 10 o 15 días.	Temperatura mínima 13°C y máxima 20°C.
Primavera	Necesita recibir una luz moderada.	Requiere de una atmósfera húmeda.	Regar cada 2 o 3 días.	Temperatura entre 16°C y 18°C.
Alstroemeria	Necesita recibir una luz moderada.	Humedad moderada.	Regar 2 o 3 veces por semana	Temperatura entre 16°C y 18°C.
Crisantemo	Necesita recibir abundante luz.	Necesita una alta humedad.	En verano regar 2 o 3 veces por semana y en invierno cada 10 días.	Temperatura entre 16°C y 18°C.

Girasol	Necesita recibir abundante luz.	Necesita una alta humedad.	En verano regar 2 o 3 veces por semana.	Temperatura entre 18°C y 20°C.
Ficus	Necesita recibir abundante luz.	Necesita una alta humedad.	En verano regar 2 o 3 veces por semana.	Temperatura mínima 13°C y máxima 24°C.

Tabla 2: Tipos de plantas y sus indicadores
Elaborado por: (Astudillo & Cadena, 2016)

(Infojardín, s.f.)

CAPÍTULO 2: ARQUITECTURA

2.1: DESCRIPCIÓN DEL PROBLEMA A SOLUCIONAR

En la actualidad existen soluciones para realizar una administración y seguimiento de jardines urbanos, sin embargo algunas de las herramientas comercializadas están enfocadas a sectores más amplios, como es el caso de cantones o ciudades. Las soluciones que se brindan para el cuidado de jardines dentro del hogar son distribuidas a un alto costo y resultan poco accesibles dentro del mercado ecuatoriano. Existe un desconocimiento y falta de interés en el mantenimiento de los jardines, es por eso que las personas que buscan tener un jardín propio en su hogar encuentran varios inconvenientes que los hacen desistir.

El tiempo que se necesita para poder iniciar un jardín dentro del hogar, resulta muy demandante para principiantes, debido a los diferentes aspectos que se debe considerar como: la preparación de la tierra, los materiales a utilizarse, adecuaciones para el lugar designado considerando condiciones climáticas, limitaciones del entorno, entre otros. Por este motivo, hoy en día se puede percibir una disminución en la creación de jardines y su fomentación.

2.2: FUNCIONALIDADES DEL PROTOTIPO

Basados en la problemática expuesta anteriormente y al análisis realizado en el marco teórico, definimos que la combinación tanto de sensores conectados a un microcontrolador, y éste a un entorno web puede solventar los inconvenientes y problemas que existen para la administración de jardines urbanos y su cuidado. Como solución se desarrollará un prototipo que cumplirá con las siguientes funcionalidades:

- **F1:** Visualizar en un entorno web los datos como: temperatura y humedad ambiental, humedad del suelo y luz.
- **F2:** Notificar alertas que representen amenazas para el mantenimiento del jardín como: escasez de agua, exceso de luz solar, posible destrucción generada por agentes externos y niveles de temperatura altos.

- **F3:** Recomendar en un entorno web sobre los cuidados para un jardín urbano como: horas en las que se debe realizar el riego, tipos de plantas que mejor se adapten al ambiente disponible, tips caseros para su cuidado.

2.3: DIAGRAMA DE ARQUITECTURA GENERAL

2.3.1: DIAGRAMA DE ARQUITECTURA

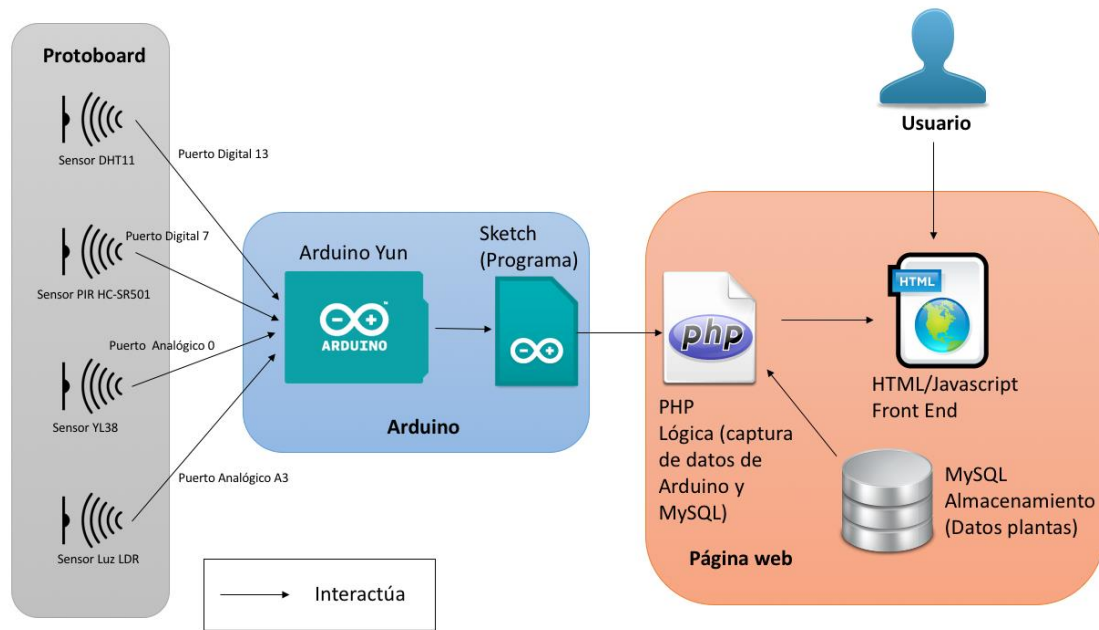


Figura 4: Diagrama de arquitectura general
Elaborado por: (Astudillo & Cadena, 2016)

En la Figura 4 se puede apreciar la arquitectura a utilizar para el desarrollo del sistema. Se puede definir que tiene tres componentes principales, que son el protoboard, arduino y la página web.

- El protoboard contiene todos los sensores que estarán ubicados dentro de la planta los cuales están encargados de recopilar los datos.
- El Arduino es el microcontrolador que funcionará a manera de servidor para poder recolectar todos los datos de los sensores y ponerlos a disposición para poder ser consultados.
- La página web es el front end que va a ser visualizado por el usuario y la parte lógica que se encarga de obtener los datos desde el Arduino y procesarlos.

2.3.2: UML DE PAQUETES

A través del diagrama de paquetes, se muestra la estructura del prototipo y la relación que existe entre sus componentes. Como se puede apreciar en la Figura 5, se tiene dos paquetes denominados Servidor Arduino y Cliente respectivamente. Cada uno de estos paquetes abarca subpaquetes que poseen una relación de dependencia entre ellos.

El paquete Servidor Arduino, contiene dos subpaquetes denominados Sensores e Interfaz, manteniendo este último una relación de dependencia con el primero. El paquete Cliente posee tres subpaquetes: Página Web que tiene una relación de dependencia con Base de datos e Interfaz Web.

Además el subpaquete Interfaz Web mantiene una relación de dependencia con el paquete Servidor Arduino debido a que los datos que se presentarán en dicha interfaz son indispensables para su correcto funcionamiento.

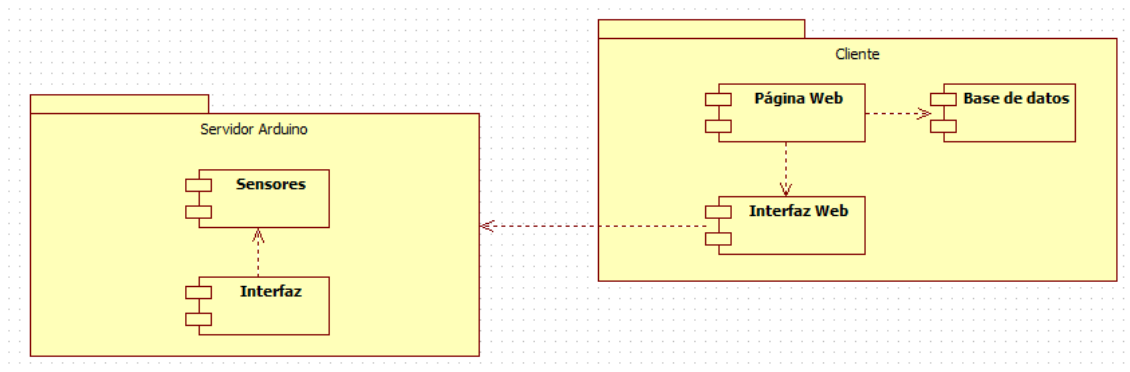


Figura 5: UML de paquetes general
Elaborado por: (Astudillo & Cadena, 2016)

2.4: DESCRIPCIÓN DE COMPONENTES DE HARDWARE

2.4.1: SENSOR DE TEMPERATURA Y HUMEDAD DHT11

2.4.1.1: Definición

El DHT11 es un sensor muy utilizado ya que mide tanto la humedad como la temperatura del ambiente. “El DHT11 es un sensor de humedad/temperatura de bajo costo y de media precisión, muy útil para los principiantes en el mundo de Arduino” (García González, 2013).

2.4.1.2: Características

Este sensor cuenta con características que lo diferencian de otros sensores que cumplen con una funcionalidad similar, y son las siguientes:

- Muy barato.
- Funciona con 3,3 y 5V de alimentación.
- Rango de temperatura: de 0° a 50° con 5% de precisión.
- Rango de humedad: de 20% al 80% con 5% de precisión.
- 1 Muestra por segundo.
- Bajo consumo.
- Devuelve la medida en °C.

(Prometec, s.f.)

Además de ser un sensor pequeño, éste es capaz de transmitir la señal hasta veinte metros de distancia, lo cual es útil para proyectos que hagan uso del mismo.

Este sensor se caracteriza por tener la señal digital calibrada por lo que asegura una alta calidad y una fiabilidad a lo largo del tiempo, ya que contiene un microcontrolador de 8 bits integrado. Está constituido por dos sensores resistivos (NTC y humedad). Tiene una excelente calidad y una respuesta rápida en las medidas. (Couto, 2012)

2.4.1.3: Ventajas

El hecho de que el DHT11 sea un sensor compuesto, es decir útil para medir la temperatura y humedad, ya se convierte en una ventaja frente a otros sensores de su gama. Según García González (2013):

El DHT11 es un sensor que proporciona una salida de datos digital. Entre sus ventajas podemos mencionar el bajo costo y el despliegue de datos digitales. Esto supone una gran ventaja frente a los sensores del tipo análogo, como el LM335 por ejemplo, en los cuales las fluctuaciones en el voltaje alteran la lectura de datos.

2.4.1.4: Conexión y lectura

El sensor DHT11 está conformado por pines en su parte inferior para realizar su debida conexión. “El sensor tiene 4 pines de conexión, el primero es VCC, el segundo es el de Señal, el tercero no se conecta y el cuarto es GND” (GeekBot, 2014).

Para la lectura de los datos que el sensor proporcione, es necesaria la descarga de una librería propia del sensor, misma que deberá incluirse dentro de la carpeta del proyecto que se realice en el ambiente de desarrollo Arduino.

2.4.2: SENSOR DE MOVIMIENTO PIR HC-SR501

2.4.2.1: Definición

El PIR HC-SR501 es un sensor que capta movimiento a través de infrarrojos.

El Sensor PIR HC-SR501 ayuda a captar movimiento, casi siempre se utiliza para detectar si un ser humano se ha movido dentro o fuera de la gama de sensores. (...) Los sensores PIR son básicamente un sensor piroeléctrico, que puede detectar los niveles de radiación infrarroja. Cualquier cosa emite cierta radiación de bajo nivel, entre más caliente esté, más radiación emite. (Nextia Fenix, s.f.)

2.4.2.2: Características

El sensor PIR HC-SR501 tiene las siguientes características:

- Sensor piroeléctrico (Pasivo) infrarrojo (También llamado PIR).
- El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización

- Rango de detección: 3 m a 7 m.
- Lente fresnel de 19 zonas, ángulo $< 100^\circ$
- Salida activa alta a 3.3 V
- Voltaje de alimentación: 4.5 VDC a 20 VDC

(Electrónicos Caldas, s.f.)

Además este sensor cuenta con dos potenciómetros y un jumper que nos dan la posibilidad de ajustar su precisión de detección, así como su tiempo de reactivación.

Los elementos de ajuste son los siguientes:

- Selector de modo: nos permite cambiar entre el modo de funcionamiento continuo o el modo de repetición. En modo continuo, si el sensor detecta movimiento de manera continuada mantendrá una señal continua. En el modo de repetición, el sensor se activará al detectar movimiento y volverá luego a su estado normal (...)
- Ajuste de sensibilidad: aumenta o disminuye la sensibilidad del sensor, con ello podemos ajustar la distancia a la que se activará y/o la cantidad de movimiento necesario para activar el sensor (...)
- Ajuste del temporizador: aumenta o disminuye el tiempo que se activará el sensor una vez detecte presencia, el rango va aproximadamente desde unos 3 segundos hasta unos 5 minutos. (Instructables, s.f.)

2.4.2.3: Conexión

El sensor PIR HC-SR501 está compuesto por pines para realizar su debida conexión.

El módulo HC-SR501 tiene 3 pines de conexión +5v, OUT (3,3v) y GND, y dos resistencias variables de calibración (Ch1 y RL2).

- Ch1: con esta resistencia podemos establecer el tiempo que se va a mantener activa la salida del sensor. (...)
- RL2: esta resistencia variable nos permite establecer la distancia de detección que puede variar entre 3-7m. (Electronilab, s.f.)

2.4.3: SENSOR DE LUZ LDR

2.4.3.1: Definición

Un LDR es aquel que permite captar la intensidad de luz, también conocido como fotorresistencia.

“El LDR (resistor dependiente de la luz) es una resistencia que varía su valor dependiendo de la cantidad de luz que la ilumina. Los valores de una fotorresistencia cuando está totalmente iluminada y cuando está totalmente a oscuras varía” (Ingeniatic, s.f.).

Este sensor de luz puede tener múltiples usos, ya sea en proyectos pequeños o grandes, brindando facilidad de conexión, al interactuar con diferentes componentes electrónicos.

2.4.3.2: Características

El LDR posee características propias, que se mencionan a continuación:

- Los valores típicos varían entre 1 M Ω , o más, en la oscuridad y 100 Ω con luz brillante.
- Disipación máxima, (50 mW-1W).
- Voltaje máximo (600V).
- Respuesta Espectral.
- El tiempo de respuesta típico de un LDR está en el orden de una décima de segundo.

(Carabal, 2011)

2.4.3.3: Conexión

Para la conexión de un LDR es necesario la presencia de una resistencia con el fin de repartir la tensión que proporciona la fuente en el mismo.

“El circuito de adaptación eléctrica de una LDR es muy sencillo, consiste en un divisor de tensión. Basta Colocar una resistencia en serie para obtener una tensión de salida analógica proporcional a la luminosidad que incide en la LDR” (Como hacer un robot, s.f.).

2.4.4: SENSOR DE HUMEDAD DEL SUELO YL 38

2.4.4.1: Definición

El sensor YL 38 es aquel que tiene la capacidad de medir la humedad del suelo al que se encuentra adherido.

El módulo YL-38 dispone de un potenciómetro con el cual se puede ajustar la sensibilidad del umbral de la salida digital para activar un relé (por ejemplo para activar una electroválvula de riego) o ser leído por un microcontrolador. Además el módulo YL-38 dispone de un LED de POWER y OUT (se activa cuando es superado el umbral) y una salida analógica que puede ser utilizada con el ADC de cualquier microcontrolador. (Educatrónica, s.f.)

2.4.4.2: Características

- Voltaje de entrada: 3.3 – 5 VCD.
- Voltaje de salida: 0 ~ 4.2 V.
- Corriente: 35 mA.
- VCC: Tensión de alimentación.
- GND: Tierra.
- A0: Salida analógica que entrega una tensión proporcional a la humedad. Puede ser medida directamente desde un puerto analógico en un microcontrolador, con Arduino, CI, etc.
- D0: Salida digital; este módulo permite ajustar cuándo el nivel lógico en esta salida pasa de bajo a alto mediante el potenciómetro.
- Dimensiones YL-38: 30 x 16 mm.

(Talos Electronics, s.f.)

2.4.5: PROTOBOARD

El ensamble de un prototipo electrónico se hace sobre un elemento llamada protoboard, en este aparato se puede montar y modificar, fácil y rápidamente, circuitos electrónicos sin necesidad de soldadura y muchas veces, sin herramientas. Una vez que el circuito en experimentación está funcionando en forma satisfactoria en el protoboard, puede ser construido de forma definitiva sobre un circuito impreso utilizando soldadura para fijar los componentes. (Cekit)

Durante el desarrollo del prototipo se utilizará un protoboard genérico para conectar los distintos circuitos porque al tratarse de un prototipo, la localización y posición de los sensores están propensos a modificación durante toda la etapa de desarrollo.

2.4.6: ARDUINO YÚN

2.4.6.1: Introducción

La creación de circuitos utilizando la tecnología VLSI (Very Large Scale Integration), permitió que a mediados de los años 80 se popularizara el uso de microcontroladores.

Los microcontroladores son computadores digitales integrados en un chip que cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada salida. El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación. (Torres, 2007)

Según la documentación de Arduino, el Arduino Yún es un microcontrolador basado en el Atmega32u4 y un microprocesador Atheros AR9331 con una distribución de Linux, llamada OpenWrt-Yún. Una de las principales características que diferencia al Arduino Yún de otras placas Arduino es que soporta tanto conectividad alámbrica como

inalámbrica, además que la comunicación con la distribución de Linux se la realiza en la misma placa.

2.4.6.2: Características

- El Arduino Yún dispone de dos interfaces de red independientes, una interfaz de Ethernet 10/100 Mbit/s y una interfaz IEEE 802.11 b/g/n, que soporta encriptación WEP, WPA y WPA2.
- Arduino Yún permite la comunicación entre los dos procesadores Atmega32U4 y el integrado en la placa Atmega32U4.
- Se dispone de la distribución de Linux OpenWrt- Yún, por lo que se tiene acceso a una versión completa de Python y la posibilidad de instalar paquetes adicionales.
- Se puede expandir la memoria de almacenamiento disponible mediante el uso de una tarjeta micro SD.
- El Arduino Yún permite la conexión de periféricos como cámaras web, ratones, teclados, entre otros haciendo uso del puerto USB Host.

(Revisión de las características hardware de la nueva Arduino Yún, n.d.)

2.4.6.3: Componentes

En la tabla 3 se puede observar las especificaciones técnicas del procesador principal incluido dentro del Arduino escogido a utilizar para el desarrollo, Arduino Yún.

Microcontrolador	Atmega32U4
Voltaje operativo	5V
Voltaje de entrada	5
Pins digitales I/O	20
Pins analógicos	12
Flash memory	32 KB
SRAM	2.5 KB
EEPROM	1 KB

Clock speed	16 MHz
-------------	--------

*Tabla 3: Microcontrolador Arduino
Elaborado por: (Astudillo & Cadena, 2015)*

En la tabla 4 se puede observar las especificaciones técnicas del otro procesador incluido dentro del microcontrolador escogido, Arduino Yún.

Procesador	Atheros AR9331
Arquitectura	MIPS 400 MHz
Voltaje Operativo	3.3 V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB	2.0 Host
Card Reader	Micro – SD only
RAM	64 MB DDR2
Clock speed	16 MHz

*Tabla 4: Microprocesador
Elaborado por: (Astudillo & Cadena, 2015)*

2.4.6.4: Análisis con otros Arduinos

	Procesador	CPU Speed	Pines digitales (I/O)	Memoria de programa (Flash)	USB	WiFi	Ethernet
Uno	ATmega328P	16 MHz	14/6	32 KB	Regular	No	No
Ethernet	ATmega328P	16 MHz	14/4	32 KB	Regular	No	Si
Mega 2560	ATmega2560	16 MHz	54/15	256 KB	Regular	No	No
Leonardo	ATmega32U4	16 MHz	20/7	32 KB	Micro	No	No
Yún	ATmega32U4 AR9331	16 MHz 400 MHz	20/7	32 KB 64 MB	Micro	Si	Si

*Tabla 5: Especificaciones de tablas Arduino
Elaborado por: (Astudillo & Cadena, 2015)*

En la Tabla 5 se muestran las placas Arduino más utilizadas, además de las características que tendrían el mayor impacto para el desarrollo de nuestro prototipo.

La velocidad de los procesadores presentes en las placas más utilizadas tiende a ser la misma, es decir 16 MHz. La principal diferencia que se puede apreciar es que la placa Arduino Yún dispone de dos procesadores integrados en la placa.

Los pines digitales para la entrada y salida de datos, serán utilizados como medio para recibir los datos de los sensores a través de la placa Arduino. La placa Arduino Mega 2560 es la que tiene la mayor cantidad de pines, siendo en el caso de este prototipo un tanto excesivo dado que se obtendrá datos de 4 sensores, por lo que los disponibles en la placa Arduino Yún son suficientes

La placa Arduino Yún es la única placa dentro de las analizadas que posee una conexión tanto alámbrica como inalámbrica, por lo que para el desarrollo de este prototipo es un punto importante a considerar. Al estar los sensores en el jardín, se utilizará una conexión inalámbrica para poder acceder a los datos y recopilarlos dentro del Arduino.

2.4.6.5: Análisis con otros microcontroladores

littleBits

Es una alternativa de microcontrolador a los Arduinos, que está caracterizado por su facilidad al momento de construir y conectar sensores con el microcontrolador.

littleBit es una plataforma de fácil uso para construir bloques electrónicos para poder inventar cualquier cosa, desde tu propio carro a control remoto, a un dispositivo para casas inteligentes. Los Bits se unen utilizando magnetos, no soldadura ni cables, además que no es necesaria la programación. (Littlebits Electronics, 2016)

littleBits es una alternativa a las placas Arduino, tiene como principal ventaja la facilidad y diversidad existente de sensores y accesorios para poder desarrollar un prototipo. Los sensores son conectados utilizando magnetos para evitar ser colocados en lugares incorrectos.

A pesar de que la sencillez es una de las principales ventajas, ésta también puede convertirse en un limitante, al momento que las partes comercializadas o disponibles no

puedan adaptarse a un ambiente determinado, que no fue considerado durante el desarrollo.

Raspberry Pi

Raspberry Pi es una mini computadora, que puede considerarse como un microcontrolador debido a su reducido tamaño y facilidad para tener entradas y salidas de datos.

The Raspberry Pi fue desarrollada por Raspberry Pi Foundation con la intención de estimular el enseñar ciencias de la computadora básica en escuelas. El diseño no incluye un disco duro o un solid state drive integrado, por lo que se basa en una tarjeta SD para el inicio y el almacenamiento. (Adafruit, 2016)

Raspberry Pi es un ordenador en un tamaño reducido, por lo que su principal ventaja frente a las placas Arduinos es que dispone de un mayor potencial para el desarrollo, porque al tener el mismo tamaño reducido dispone de distribuciones de Linux completas y no versiones adaptadas. De modo que las funcionalidades que se pueden desarrollar, pueden tener un mayor grado de complejidad al que se puede desarrollar con las placas Arduino, limitadas por sus especificaciones de hardware.

2.5: DESCRIPCIÓN COMPONENTES DE SOFTWARE

2.5.1: ENTORNO ARDUINO

“El entorno de desarrollo integrado Arduino, o Arduino Software (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús” (Arduino, 2015).

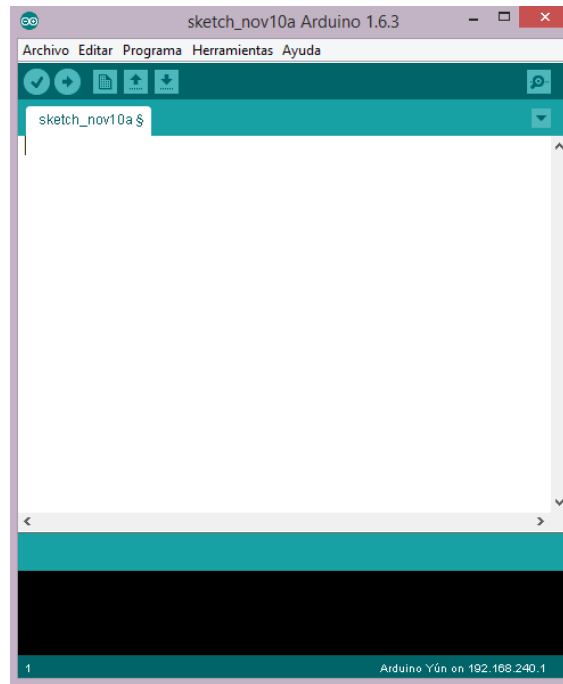


Figura 6: Entorno de Arduino
Fuente: (Astudillo & Cadena, 2015)

Dentro de este editor se escriben las líneas de código que darán como resultado un programa, o mejor conocido como “sketch” o proyecto, el mismo que deberá ser precargado al microcontrolador de la placa Arduino. Para realizar dicha función, Arduino ofrece un programa pequeño llamado bootloader, que permite cargar el código deseado a la placa sin tener que utilizar un hardware adicional. Después de precargar las líneas de código, se inicia el programa previamente escrito y desarrollado. (Letrán, 2011)

2.5.1.1: ESTRUCTURA DE UN PROGRAMA

Existen dos partes principales que constan en la estructura básica en cuanto al lenguaje de programación utilizado en Arduino. Mejor conocidas como funciones, éstas incluyen tanto declaraciones de variables como instrucciones propias para cada tipo de programa.

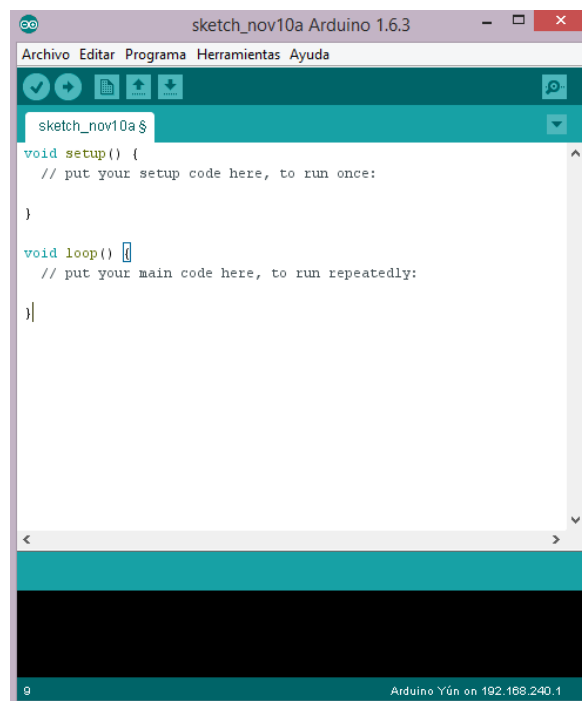
Setup()

La función `setup()` o de configuración, es aquella que se ejecuta una sola vez cuando el programa empieza. Aquí es donde se declaran las variables necesarias y donde se

inicializa el modo de trabajo de los pins de la placa, así como la definición del estado inicial de las salidas de la placa (Arduino, s.f.).

Loop()

La función loop() o bucle, es aquella que tendrá como función ejecutarse continuamente o de forma cíclica. Aquí es donde se escribirán las lecturas de entrada, activación de salidas, entre otros, con el fin de responder ante los eventos que la placa produzca en ese instante (Arduino, s.f.).



*Figura 7: Estructura de un programa en Arduino
Fuente: (Astudillo & Cadena, 2015)*

CAPÍTULO 3: PRIMERA ITERACIÓN ARDUINO Y SENSORES

En el presente capítulo se definirá los requerimientos funcionales y no funcionales a considerarse para el desarrollo durante la primera iteración. Se define el flujo para la lectura de los datos de los diferentes sensores a utilizarse.

3.1: ANÁLISIS

Los sensores y su interacción con el microcontrolador Arduino Yún, debe contener permitir realizar funcionalidades específicas que brinden los datos necesarios para poder ser enviados al microcontrolador Arduino Yún.

3.1.1: DESCRIPCIÓN DE REQUERIMIENTOS

3.1.1.1: REQUERIMIENTOS FUNCIONALES

RF1: El microcontrolador Arduino Yún permitirá la consulta de los indicadores ambientales.

RF2: El microcontrolador Arduino Yún permitirá la consulta del indicador de las plantas.

RF3: El microcontrolador Arduino Yún permitirá detectar el movimiento cercano a las plantas.

3.1.1.2: REQUERIMIENTOS NO FUNCIONALES

RNF1: Los sensores a utilizar tienen que ser compatibles con la plataforma Arduino Yún.

RNF2: La programación del Arduino Yún debe ser implementada en Processing/Wiring.

3.1.2: CASOS DE USO

3.1.2.1: DIAGRAMA GENERAL PRIMERA ITERACIÓN

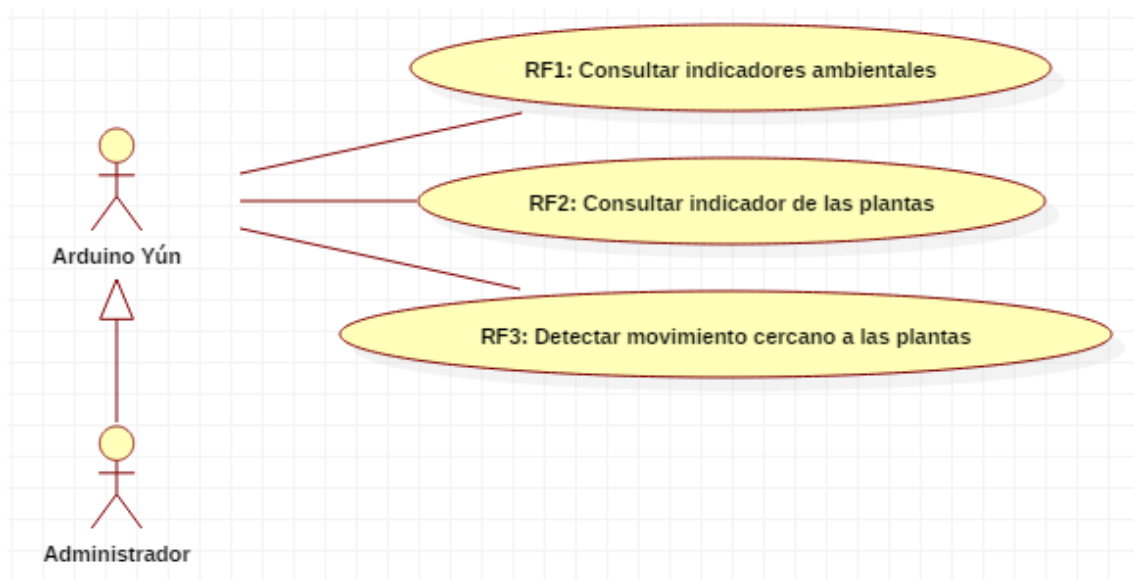


Figura 8: Diagrama general primera iteración
Elaborado por: (Astudillo & Cadena, 2016)

- Siguiendo nivel de **RF1**:

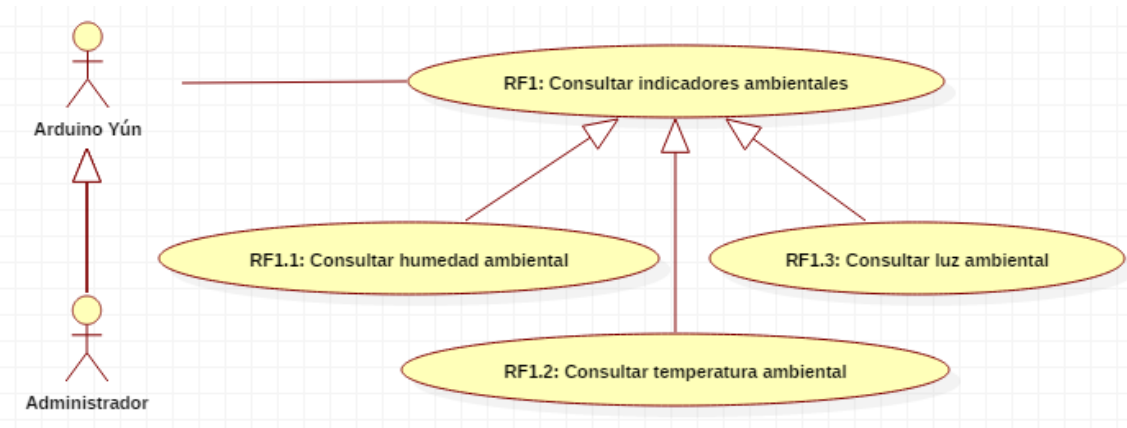


Figura 9: Siguiete nivel RF1
Elaborado por: (Astudillo & Cadena, 2016)

– Siguiete nivel **RF2**:

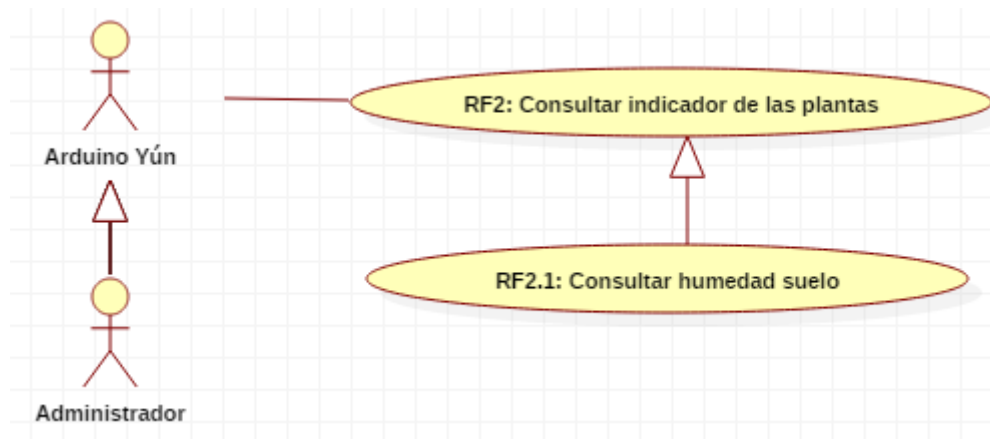


Figura 10: Siguiete nivel RF2
Elaborado por: (Astudillo & Cadena, 2016)

– Siguiete nivel **RF3**:

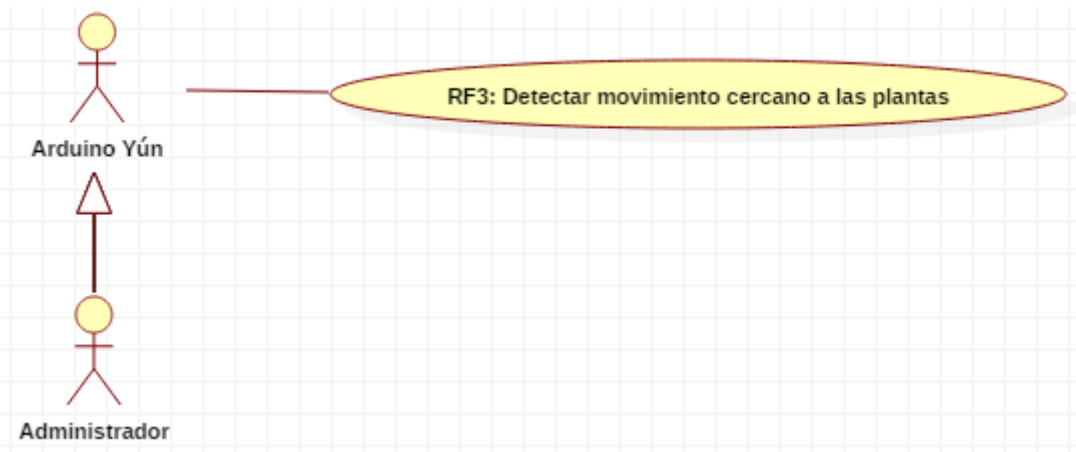


Figura 11: Siguiendo nivel RF3
Elaborado por: (Astudillo & Cadena, 2016)

3.1.2.2: DIAGRAMAS A DETALLE

– RF1.1: Consultar humedad ambiental

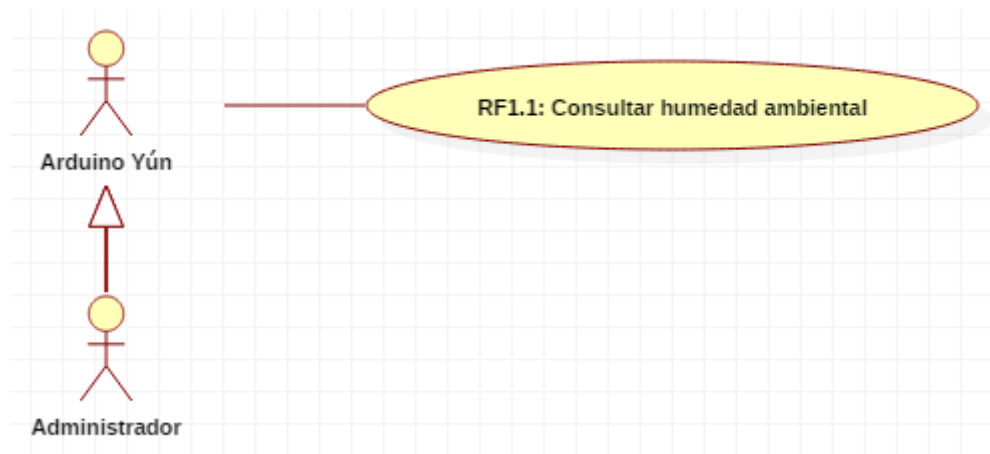


Figura 12: Diagrama consultar humedad ambiental
Elaborado por: (Astudillo & Cadena, 2016)

Actores: Arduino Yún, Administrador

Flujo principal:

1. El administrador verifica el sketch que contiene consultar humedad ambiental, presionando el botón verificar en el IDE de Arduino Yún.
2. El IDE de Arduino Yún compila el sketch seleccionado. (E1)

3. El administrador sube el sketch que contiene consultar humedad ambiental. (E2)
4. Arduino Yún carga el sketch.
5. El administrador presiona el botón Monitor Serie del IDE de Arduino Yún.
6. Arduino Yún lee los datos obtenidos del sensor de humedad ambiental.
7. El administrador visualiza los datos proporcionados por el Arduino Yún en el Monitor Serie.

Flujo alterno:

Si existe un cambio de puerto del Arduino Yún, pulsar el botón Reset de la placa y volver a subir el sketch.

Excepciones:

E1

Causa: Existen errores de sintaxis en el sketch.

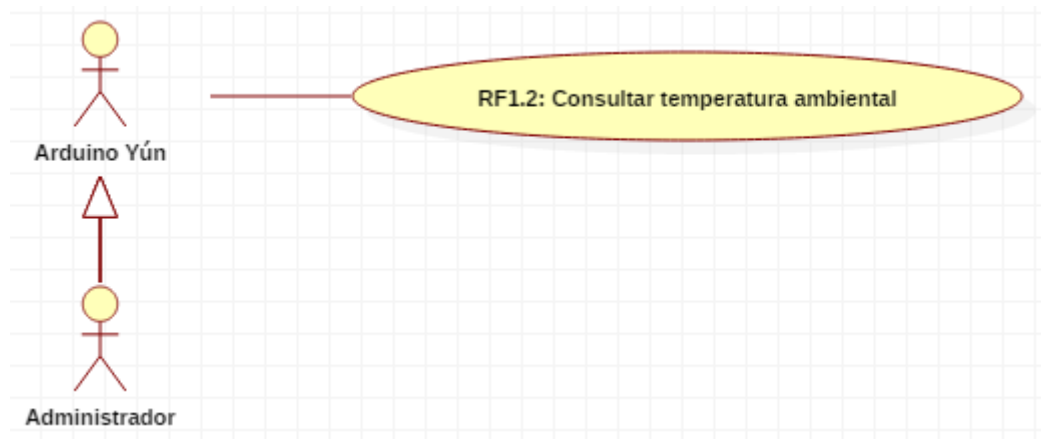
Posible solución: Detectar los errores y repararlos.

E2

Causa: No se encuentra la placa Arduino en el puerto seleccionado.

Posible solución: Comprobar si se ha seleccionado el puerto correcto.

– RF1.2: Consultar temperatura ambiental



*Figura 13: Diagrama consultar temperatura ambiental
Elaborado por: (Astudillo & Cadena, 2016)*

Actores: Arduino Yún, Administrador

Flujo principal:

1. El administrador verifica el sketch que contiene consultar temperatura ambiental, presionando el botón verificar en el IDE de Arduino Yún.
2. El IDE de Arduino Yún compila el sketch seleccionado. (E1)
3. El administrador sube el sketch que contiene consultar temperatura ambiental. (E2)
4. Arduino Yún carga el sketch.
5. El administrador presiona el botón Monitor Serie del IDE de Arduino Yún.
6. Arduino Yún lee los datos obtenidos del sensor de temperatura ambiental.
7. El administrador visualiza los datos proporcionados por el Arduino Yún en el Monitor Serie.

Flujo alterno:

Si existe un cambio de puerto del Arduino Yún, pulsar el botón Reset de la placa y volver a subir el sketch.

Excepciones:

E1

Causa: Existen errores de sintaxis en el sketch.

Posible solución: Detectar los errores y repararlos.

E2

Causa: No se encuentra la placa Arduino en el puerto seleccionado.

Posible solución: Comprobar si se ha seleccionado el puerto correcto.

– RF1.3: Consultar luz ambiental

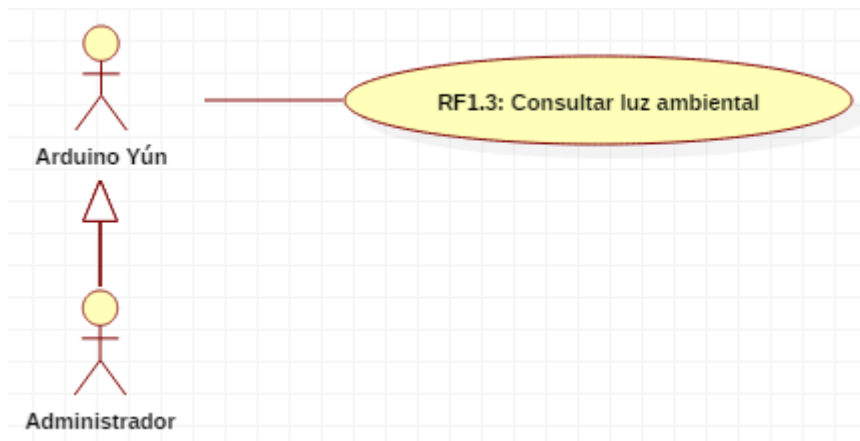


Figura 14: Diagrama consultar luz ambiental
Elaborado por: (Astudillo & Cadena, 2016)

Actores: Arduino Yún, Administrador

Flujo principal:

8. El administrador verifica el sketch que contiene consultar luz ambiental, presionando el botón verificar en el IDE de Arduino Yún.
9. El IDE de Arduino Yún compila el sketch seleccionado. (E1)
10. El administrador sube el sketch que contiene consultar luz ambiental. (E2)
11. Arduino Yún carga el sketch.
12. El administrador presiona el botón Monitor Serie del IDE de Arduino Yún.

13. Arduino Yún lee los datos obtenidos del sensor de luz ambiental.
14. El administrador visualiza los datos proporcionados por el Arduino Yún en el Monitor Serie.

Flujo alterno:

Si existe un cambio de puerto del Arduino Yún, pulsar el botón Reset de la placa y volver a subir el sketch.

Excepciones:

E1

Causa: Existen errores de sintaxis en el sketch.

Posible solución: Detectar los errores y repararlos.

E2

Causa: No se encuentra la placa Arduino en el puerto seleccionado.

Posible solución: Comprobar si se ha seleccionado el puerto correcto.

– RF2.1: Consultar humedad suelo

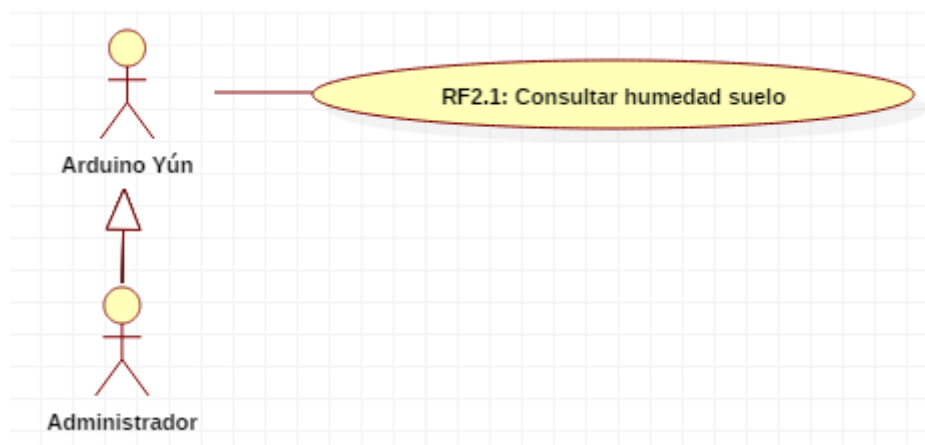


Figura 15: Diagrama consultar humedad suelo
Elaborado por: (Astudillo & Cadena, 2016)

Actores: Arduino Yún, Administrador

Flujo principal:

1. El administrador verifica el sketch que contiene consultar humedad del suelo, presionando el botón verificar en el IDE de Arduino Yún.
2. El IDE de Arduino Yún compila el sketch seleccionado. (E1)
3. El administrador sube el sketch que contiene consultar humedad del suelo. (E2)
4. Arduino Yún carga el sketch.
5. El administrador presiona el botón Monitor Serie del IDE de Arduino Yún.
6. Arduino Yún lee los datos obtenidos del sensor de humedad de suelo.
7. El administrador visualiza los datos proporcionados por el Arduino Yún en el Monitor Serie.

Flujo alterno:

Si existe un cambio de puerto del Arduino Yún, pulsar el botón Reset de la placa y volver a subir el sketch.

Excepciones:

E1

Causa: Existen errores de sintaxis en el sketch.

Posible solución: Detectar los errores y repararlos.

E2

Causa: No se encuentra la placa Arduino en el puerto seleccionado.

Posible solución: Comprobar si se ha seleccionado el puerto correcto.

3.2: DISEÑO

Para un mejor entendimiento del diseño para la primera iteración, se presenta el diagrama de arquitectura del mismo, con los componentes necesarios que interactuarán con el microcontrolador Arduino Yún.

3.2.1: DIAGRAMA DE ARQUITECTURA

La primera iteración se centra en el diseñar y definir la conexión desde los sensores a utilizarse al microcontrolador Arduino Yún. Los componentes que serán parte de esta iteración son el protoboard y el Arduino Yún.

3.2.1.1: DIAGRAMA A DETALLE PRIMERA ITERACIÓN

Dentro de la Figura 16 se puede observar cómo se va a realizar el flujo de datos desde el protoboard hacia el microcontrolador Arduino Yún. Se definen los puertos mediante los cuales se enviarán los datos.

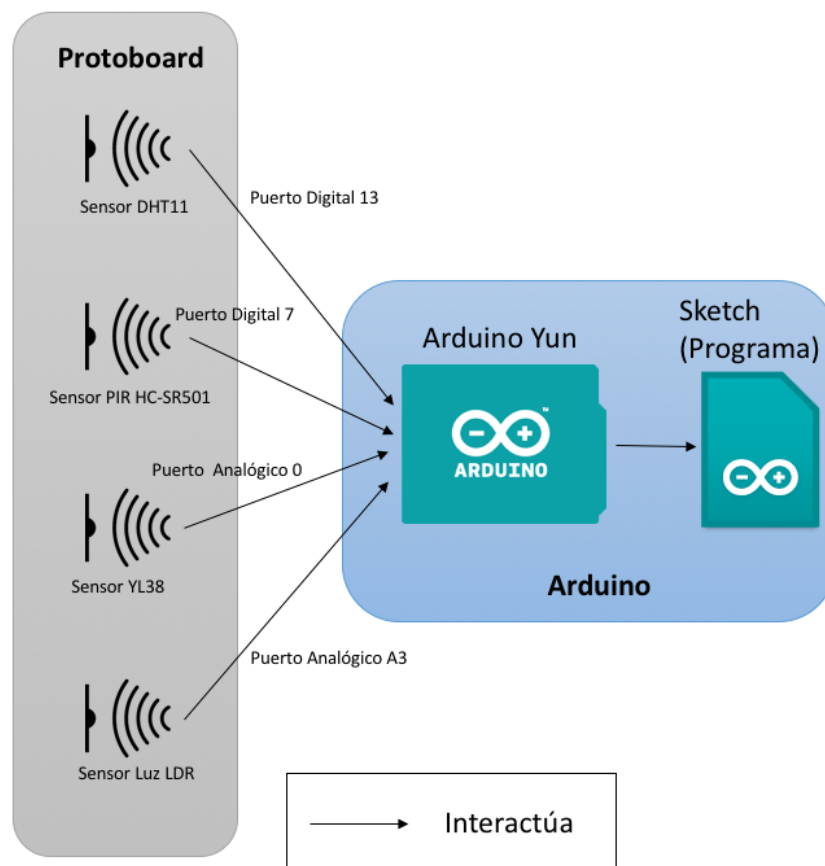


Figura 16: Diagrama a detalle primera iteración
Elaborado por: (Astudillo & Cadena, 2016)

3.3: CODIFICACIÓN

El programa que se está utilizando para el microcontrolador Arduino fue desarrollado con el lenguaje de programación Wiring, donde se genera un archivo denominado en la plataforma de Arduino como sketch.

Dentro del sketch se definen si los puertos del Arduino van a ser puertos de entrada o puertos de salida.

```
pinMode(13, INPUT); //sensor humedad y temperatura
pinMode(A0, INPUT); // sensor de suelo
pinMode(A3, INPUT); //sensor de luz
pinMode(7, INPUT); //sensor movimiento
pinMode(11, OUTPUT); //led
```

Las placas Arduino se caracterizan por tener dos tipos de pines, se dispone pines digitales y pines analógicos. En este caso se utilizará el puerto digital 11 como un puerto de salida para poder mostrar cuando ha sido detectado un movimiento.

Se utilizarán dos puestos analógicos para poder facilitar la lectura de la señal enviada por los sensores de humedad de suelos y de foto receptor de luz.

Dentro de los sketches de Arduino es posible definir una función llamada loop que se ejecuta a lo largo del tiempo. Dentro de esta función nos basamos para el uso de los objetos ya definidos Client y Server para permitir que la placa Arduino funcione como un servidor.

```
void loop() {
  YunClient client = server.accept();
  if (client) {
    process(client);
  }
}
```

```

    client.stop();
}

```

Dentro de la función loop también se llama a la función process que se encarga de presentar los datos que se obtienen de los diferentes sensores conectados a través de los puerto, que posteriormente van a ser presentados a los usuarios.

```

temperatura = dht.readTemperature();
humedad = dht.readHumidity();
suelo = analogRead(A0);
luz = analogRead(A3);

```

La función AnalogRead permite poder leer el voltaje enviado por el sensor, en donde el pin es capaz de convertir de 0 a 5V en valores de entre 0 a 1023 respectivamente, que de acuerdo a que sensor es el que registra ese valor se interpretara.

Las funciones utilizadas readTemperature y readHumidity son funciones específicas para el sensor DHT11.

3.4: PRUEBAS

3.4.1: Sensor de humedad y temperatura

Acción	Consultar la temperatura en la ruta <i>IPArduino/arduino/temperatura/13</i> , cuando el sensor DTH11 esté conectado al Arduino con el sketch que contiene la función para leer la temperatura
Resultado Esperado	Se puede visualizar la temperatura ambiental desde el navegador de un dispositivo
Evidencia	

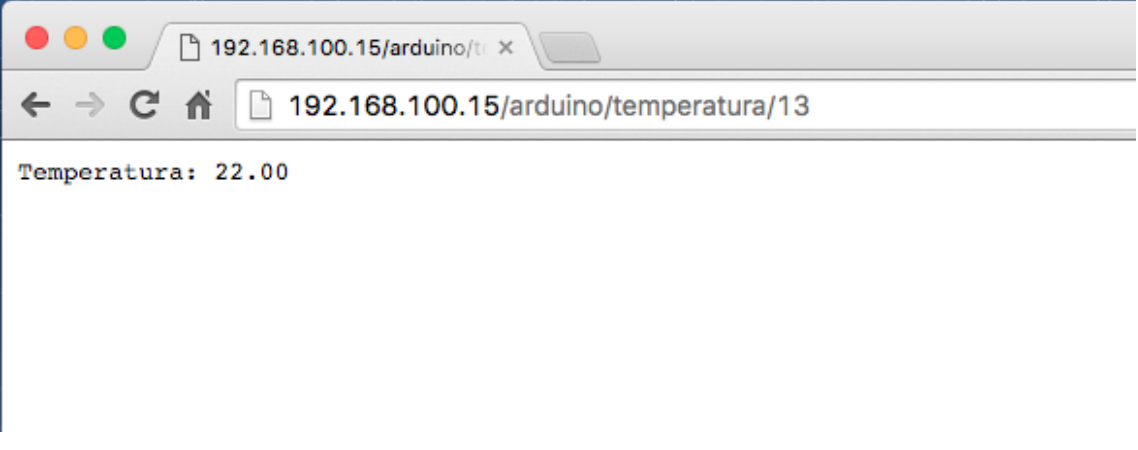
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 6: Primera prueba sensor de humedad y temperatura
Elaborado por: (Astudillo & Cadena, 2016)

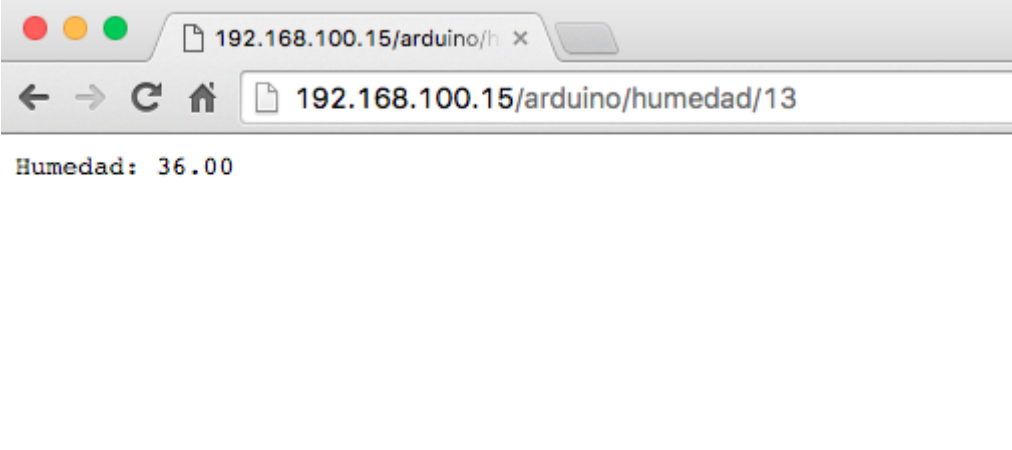
Acción	Consultar la humedad en la ruta <i>IPArduino/arduino/humedad/13</i> , cuando el sensor DTH11 esté conectado al Arduino con el sketch que contiene la función para leer la temperatura
Resultado Esperado	Se puede visualizar la humedad ambiental desde el navegador de un dispositivo
Evidencia	
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 7: Segunda prueba sensor de humedad y temperatura
Elaborado por: (Astudillo & Cadena, 2016)

3.4.2: Sensor de luz

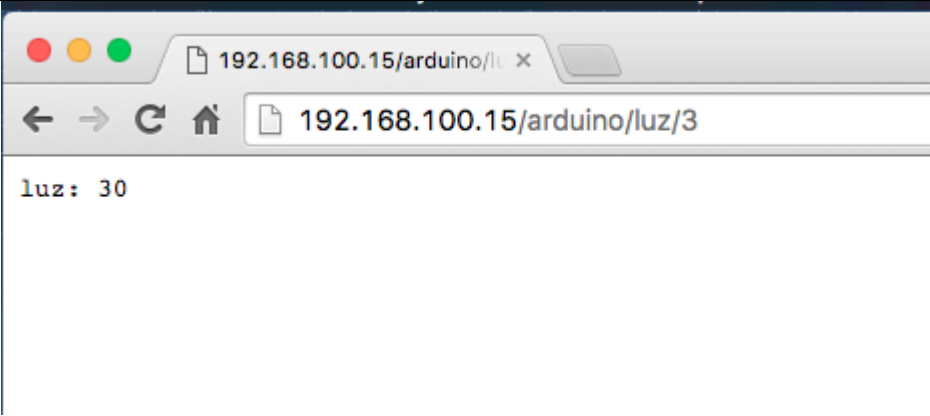
Acción	Consultar la humedad en la ruta <i>IPArduino/arduino/humedad/13</i> , cuando esté conectado al Arduino con el sketch que contiene la función para leer la luz
Resultado Esperado	Se puede visualizar la luz solar desde el navegador de un dispositivo
Evidencia	
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 8: Prueba sensor de luz
Elaborado por: (Astudillo & Cadena, 2016)

CAPÍTULO 4: SEGUNDA ITERACIÓN PÁGINA WEB

En el presente capítulo se definirán tanto los requerimientos funcionales como no funcionales para el diseño de la página web. Para cada requerimiento funcional se especificará su comportamiento mediante diagramas de casos de uso, y una vez detallado el proceso de cada uno, se presentará la codificación y las pruebas de los mismos que comprueben su correcto funcionamiento.

4.1: ANÁLISIS

La página web que es parte del prototipo, debe contener funcionalidades específicas que brinden al usuario facilidad al momento de administrar su jardín. A continuación se describen los requerimientos funcionales y no funcionales que constarán en la página web.

4.1.1: DESCRIPCIÓN DE REQUERIMIENTOS

4.1.1.1: REQUERIMIENTOS FUNCIONALES

RF4: El usuario será capaz de visualizar el diseño en una página web para los indicadores de cuidado de las plantas.

RF5: El usuario será capaz de visualizar el diseño en una página web para las alertas de precauciones a tomar en cuenta para el cuidado de las plantas.

RF6: El usuario será capaz de visualizar el diseño en una página web para las recomendaciones para sugerencia de plantas.

RF7: El usuario será capaz de visualizar el diseño en una página web para los tips caseros para el cuidado de las plantas.

4.1.1.2: REQUERIMIENTOS NO FUNCIONALES

RNF3: La página web debe ser desarrollada utilizando PHP, CSS y HTML.

4.1.2: CASOS DE USO

4.1.2.1: DIAGRAMA GENERAL SEGUNDA ITERACIÓN

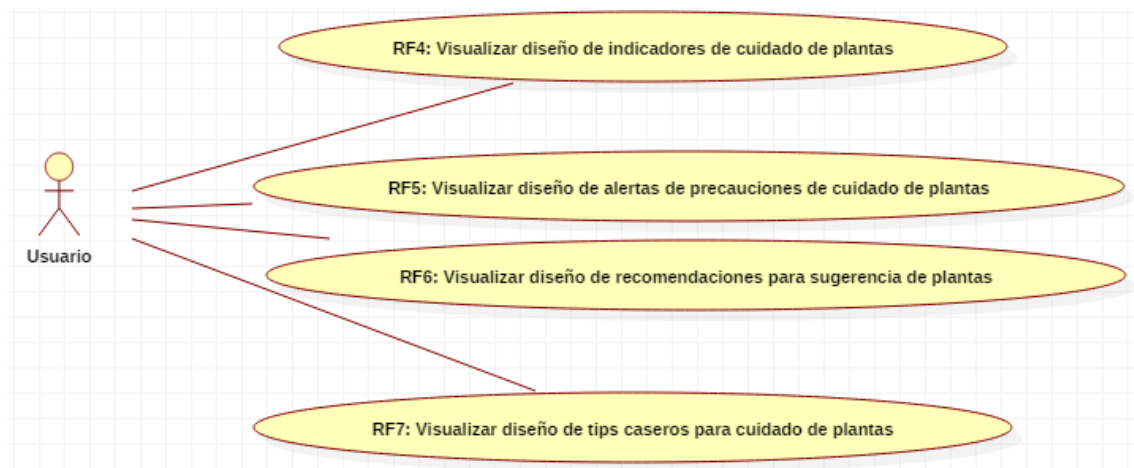


Figura 17: Diagrama general segunda iteración
Elaborado por: (Astudillo & Cadena, 2016)

4.1.2.2: DIAGRAMAS A DETALLE

– RF4: Visualizar diseño de indicadores de cuidado de plantas

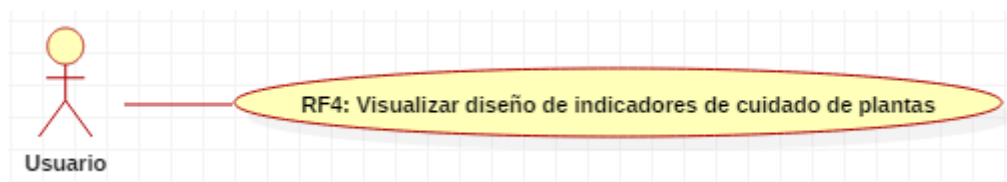


Figura 18: Diagrama visualizar diseño de indicadores de cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web. (E1)
2. El usuario presiona en la opción Inicio del menú principal.
3. El usuario visualiza el diseño para los indicadores de cuidado de las plantas.

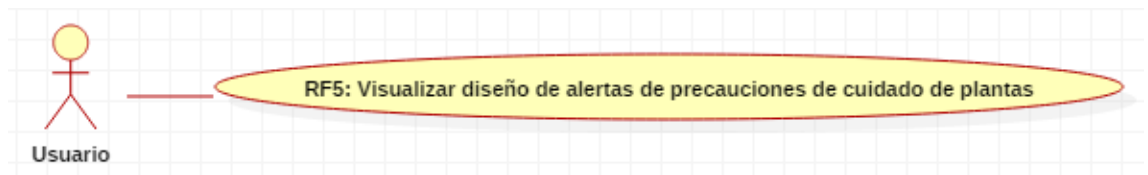
Excepciones:

E1

Causa: Existe un error con el servidor PHP.

Posible solución: Reiniciar el servidor.

– **RF5: Visualizar diseño de alertas de precauciones de cuidado de plantas**



*Figura 19: Diagrama visualizar diseño de alertas de precauciones de cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)*

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web. (E1)
2. El usuario visualiza el diseño de mensaje de alerta.

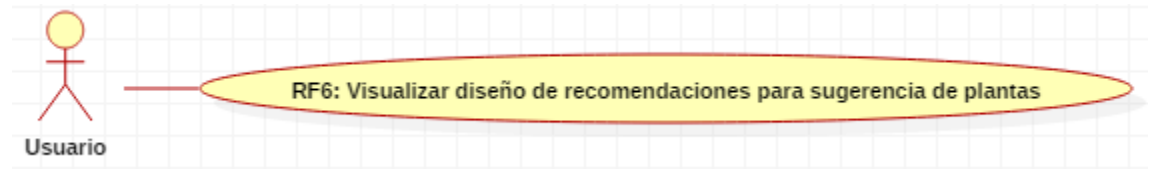
Excepciones:

E1

Causa: Existe un error con el servidor PHP.

Posible solución: Reiniciar el servidor..

– **RF6: Visualizar diseño de recomendaciones para sugerencia de plantas**



*Figura 20: Visualizar diseño de recomendaciones para sugerencia de plantas
Elaborado por: (Astudillo & Cadena, 2016)*

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web. (E1)
2. El usuario selecciona la opción Recomendaciones del menú principal para la sugerencia de plantas.
3. El usuario visualiza el diseño de las recomendaciones para el cuidado de las plantas.

Excepciones:

E1

Causa: Existe un error con el servidor PHP.

Posible solución: Reiniciar el servidor.

– **RF7: Visualizar diseño de tips caseros para cuidado de plantas**

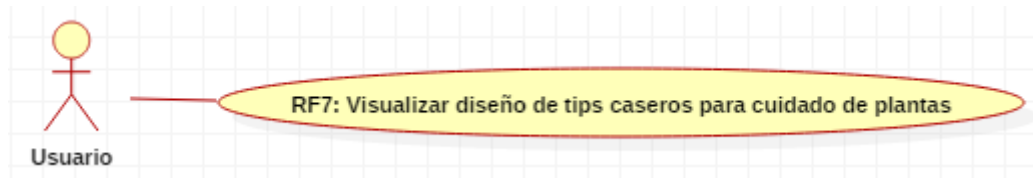


Figura 21: Diagrama visualizar diseño de tips caseros para cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web. (E1)
2. El usuario selecciona la opción Tips del menú principal.
3. El usuario visualiza el diseño de los tips caseros para el cuidado de las plantas.

Excepciones:

E1

Causa: Existe un error con el servidor PHP.

Posible solución: Reiniciar el servidor.

4.2: DISEÑO

Para un mejor entendimiento del diseño para la segunda iteración, se presenta el diagrama de arquitectura del mismo, con los componentes necesarios que interactuarán en la página web.

4.2.1: DIAGRAMA DE ARQUITECTURA

Con el fin de mostrar una visión simplificada de la página web, se presentará a continuación el diagrama de arquitectura.

4.2.1.1: DIAGRAMA A DETALLE SEGUNDA ITERACIÓN

La segunda iteración se centra en el diseño y construcción de la página web. Los componentes que serán parte de esta iteración son el usuario y los archivos HTML codificados para su fin, ambos relacionados directamente como se puede observar en la Figura 22. El usuario será quien manipule la página web que pondrá a su disposición las funcionalidades definidas previamente como requerimientos.

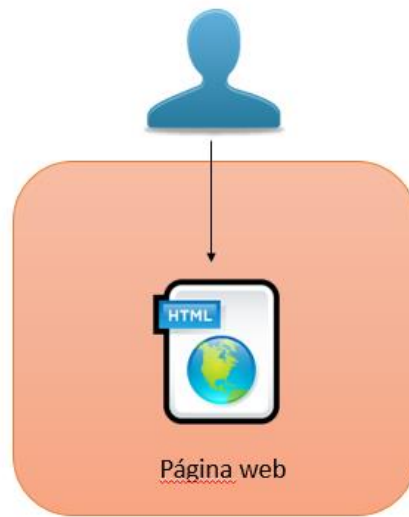


Figura 22: Diagrama a detalle segunda iteración
Elaborado por: (Astudillo & Cadena, 2016)

4.3: CODIFICACIÓN

La página web ha sido desarrollada en un lenguaje HTML, y para el aspecto de la misma se ha aplicado el lenguaje CSS.

Para el fin del prototipo propuesto, la página web está compuesta por un menú principal con tres opciones: inicio, recomendaciones y tips. Cada opción tiene su re direccionamiento a su propio archivo php.

```
<div id="navigation_wrapper">
  <ul id="navigation">
    <li><a href="inicio.php" title="Inicio">Inicio</a></li>
    <li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
    <li><a href="tips.php" title="Tips">Tips</a></li>
  </ul>
</div>
```

Además, cada una de las opciones incluye los indicadores de cuidado de plantas para la apreciación de sus valores respectivos y para su actualización continua.

Indicador temperatura ambiental:

```
<div class="banner">
  <h2>Temperatura <br /> Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form2" name="form2" method="post" action="">
      <p>
        <label for="temperaturaAmbiental"></label>
        <input type="text" class="textbox" name="temperaturaAmbiental"
id="temperaturaAmbiental" readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador1"></label>
    </form>
  </div>
</div>
```


El propósito que tiene la actualización continua de los indicadores de cuidado de plantas, es la visualización de una alerta que dependerá si los valores de los indicadores se encuentran o no dentro del rango definido.

Para el aspecto de la alerta se utiliza la herramienta SweetAlert. Una vez descargada, se la incluye dentro de la carpeta del proyecto, para luego en cada una las opciones de la página web importar la librería y el archivo CSS de la herramienta. Con un poco de código JavaScript es posible el funcionamiento de la alerta.

```
<link rel="stylesheet" type="text/css" href="sweetalert/dist/sweetalert.css">
<script src="sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript">
function checkCookies() {
  sweetAlert('Cuidado!', 'Por favor, revisa tus indicadores');
  window.onload = checkCookies;
}</script>
```

4.3.1: Inicio.php

La opción Inicio es la encargada de mostrar los indicadores de cuidado de plantas tal y como se mencionó en el apartado anterior.

4.3.2: Recomendaciones.php

La opción Recomendaciones, además de mostrar los indicadores de cuidado de plantas, incluye un pequeño cuestionario junto con un botón, con el fin de que posteriormente se le de funcionalidad al mismo, para la sugerencia de plantas de acuerdo a las respuestas respectivas.

```
<form class="form" id="form1" name="form1" method="post"
action="recomendacionFinal.php">

  <p>
    <label class="textPregunta" >¿Espacio disponible para su jardín?</label>
  </p> <p>&nbsp;</p>

  <p>
    <label class="radioPregunta">
      <input type="radio" name="radioPreguntaUno" value="árbol"
id="radioPreguntaUno_0" />
      Amplio </label>
      <label class="radioPregunta">
        <input type="radio" name="radioPreguntaUno" value="arbusto"
id="radioPreguntaDos_0" />
        Moderado </label>
        <label class="radioPregunta">
          <input type="radio" name="radioPreguntaUno" value="herbácea"
id="radioPreguntaTres_1" />
          Limitado</label>
        </p>
      <p>
        <input type="submit" class="button" name="botonSubmitRecomendaciones"
id="botonSubmitRecomendaciones" value="Recomendar" />
        </p> <p> <br /> </p>
    </form>
```

4.3.3: *Tips.php*

La opción Tips, además de mostrar los indicadores de cuidado de plantas, incluye imágenes diseñadas donde se presentan tips para el cuidado de plantas.

```
<p></p>
```

4.4: PRUEBAS

4.4.1: Diseño indicadores de cuidados de plantas

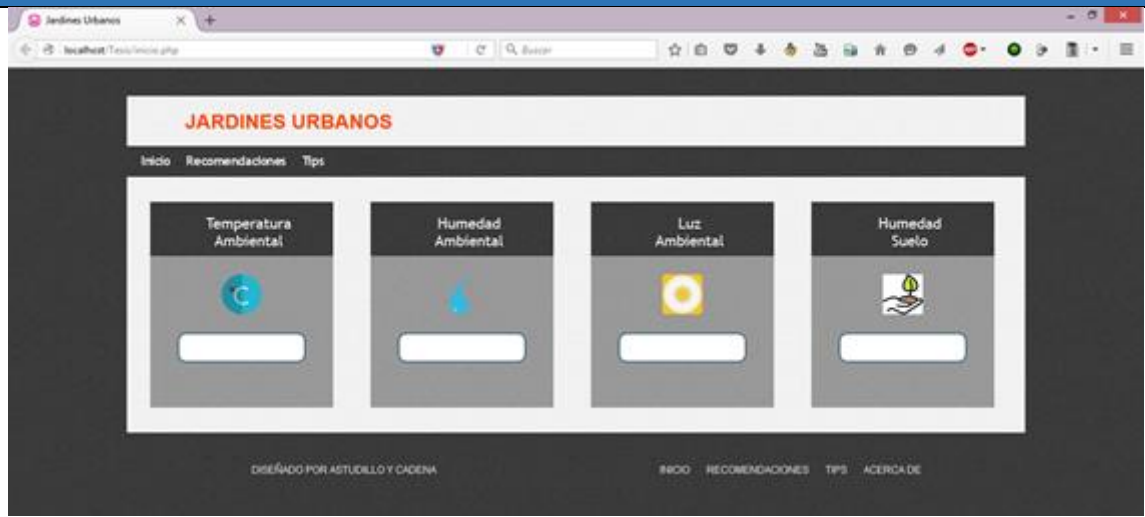
Acción	Ingresa a la página web, dirigirse a la opción Inicio del menú principal y visualizar el diseño donde se presentarán los indicadores de cuidados de plantas como: temperatura y humedad ambiental, luz ambiental y humedad del suelo.
Resultado Esperado	Se puede visualizar el diseño previsto para los indicadores de cuidado de plantas desde el navegador de un dispositivo.
Evidencia	
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 9: Prueba diseño indicadores de cuidados de plantas
Elaborado por: (Astudillo & Cadena, 2016)

4.4.2: Diseño alertas de precauciones de cuidados de plantas

4.4.2.1: Diseño mensajes de alerta

Acción	Ingresa a la página web y visualizar el diseño donde se presentarán los mensajes de alerta de cuidados de plantas.
--------	--

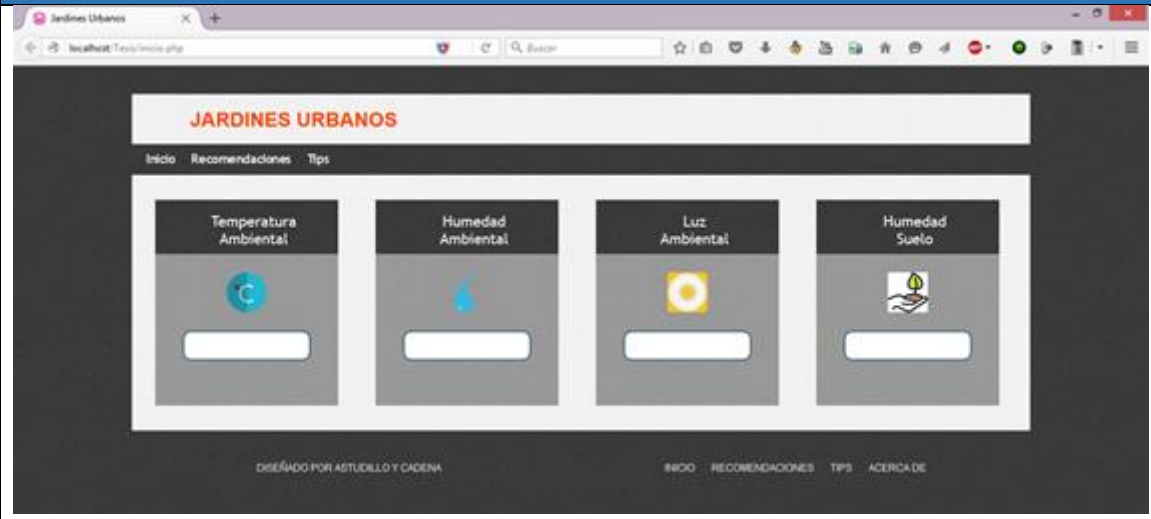
Resultado Esperado	Se puede visualizar el diseño previsto para los mensajes de alerta de cuidado de plantas desde el navegador de un dispositivo.
Evidencia	
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 10: Prueba diseño mensajes de alerta
Elaborado por: (Astudillo & Cadena, 2016)

4.4.2.2: Diseño alerta

Acción	Ingresar a la página web y visualizar el diseño donde se presentará la alerta de cuidados de plantas.
Resultado Esperado	Se puede visualizar el diseño previsto para la alerta de cuidado de plantas desde el navegador de un dispositivo.
Evidencia	

	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 11: Prueba diseño alerta
Elaborado por: (Astudillo & Cadena, 2016)

4.4.3: Diseño recomendaciones para sugerencia de plantas

Acción	Ingresar a la página web, dirigirse a la opción Recomendaciones del menú principal y visualizar el diseño donde se presentarán las recomendaciones para sugerencia de plantas.
Resultado Esperado	Se puede visualizar el diseño previsto para las recomendaciones de sugerencia de plantas desde el navegador de un dispositivo.
Evidencia	

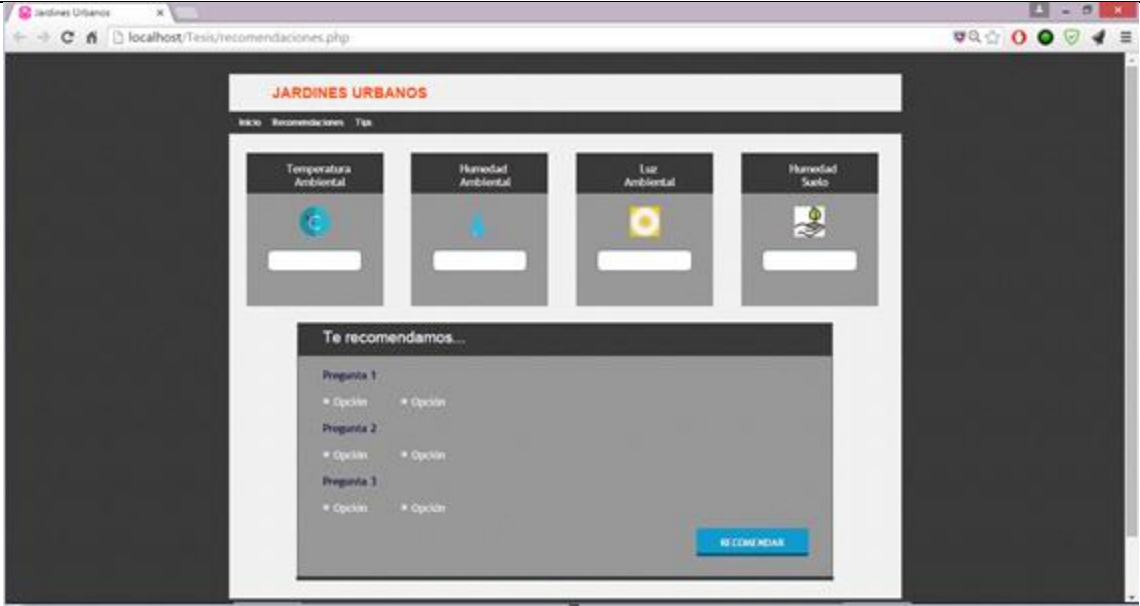
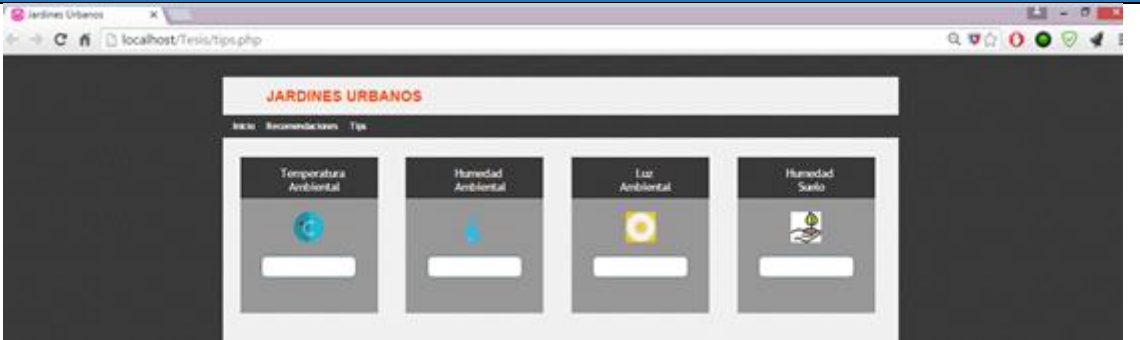
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 12: Prueba diseño recomendaciones para sugerencia de plantas
Elaborado por: (Astudillo & Cadena, 2016)

4.4.4: Diseño tips caseros para cuidado de plantas

Acción	Ingresar a la página web, dirigirse a la opción Tips del menú principal y visualizar el diseño donde se presentarán los tips para cuidado de plantas.
Resultado Esperado	Se puede visualizar el diseño previsto para los tips para cuidado de plantas desde el navegador de un dispositivo.
Evidencia	
	

Resultado	Se pasó la prueba satisfactoriamente	

Tabla 13: Prueba diseño tips caseros para cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)

CAPÍTULO 5: INTEGRACIÓN PÁGINA WEB Y ARDUINO

En el presente capítulo se describirán requerimientos funcionales y no funcionales que reflejen el completo funcionamiento del prototipo, es decir, compactando los requerimientos de la primera y segunda iteración. Se especificará el comportamiento de cada requerimiento funcional mediante diagramas de casos de uso, y una vez detallado

el proceso de cada uno, se presentará la codificación y las pruebas de los mismos que comprueben que se han implementado correctamente.

5.1: ANÁLISIS

Para que el prototipo sea completamente eficaz en su última fase, debe cumplir con funcionalidades que integren los datos obtenidos de la primera iteración, así como el uso de la página web desarrollada en la segunda iteración. A continuación se describen los requerimientos funcionales y no funcionales para dicha integración.

5.1.1: DESCRIPCIÓN DE REQUERIMIENTOS

5.1.1.1: REQUERIMIENTOS FUNCIONALES

RF8: El usuario será capaz de visualizar los datos procesados de los sensores dentro del área designada para los indicadores de cuidado de plantas de la página web.

RF9: El usuario será capaz de visualizar alertas de precauciones que dependan del estado de los indicadores de cuidado de las plantas.

RF10: El usuario será capaz de visualizar recomendaciones para sugerencia de plantas de acuerdo al cuestionario presentado en la página web.

5.1.1.2: REQUERIMIENTOS NO FUNCIONALES

RNF4: La base de datos debe ser desarrollada utilizando MySQL.

5.1.2: CASOS DE USO

5.1.2.1: DIAGRAMA GENERAL TERCERA ITERACIÓN

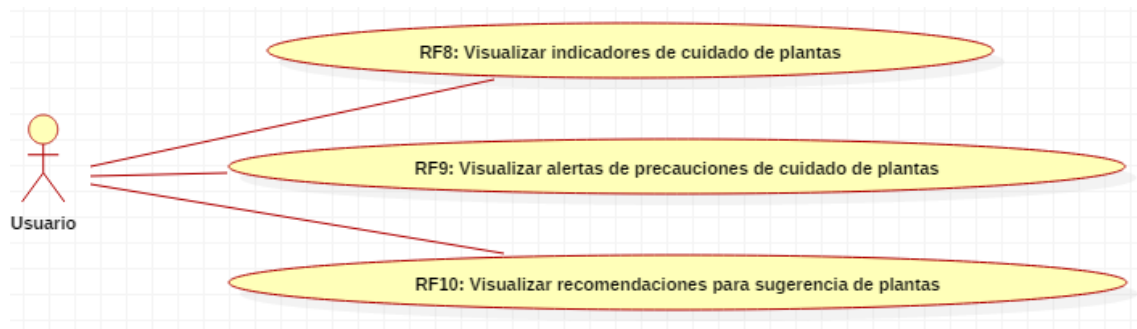


Figura 23: Diagrama general tercera iteración
Elaborado por: (Astudillo & Cadena, 2016)

5.1.2.2: DIAGRAMAS A DETALLE

– RF8: Visualizar indicadores de cuidado de plantas

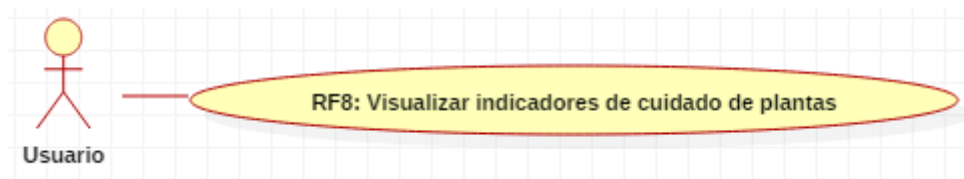


Figura 24: Diagrama visualizar indicadores de cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web.
2. El usuario presiona en la opción Inicio del menú principal.
3. El sistema genera los indicadores de cuidado de las plantas. (E1)
4. El usuario visualiza en la página web los indicadores de cuidado de las plantas.

Excepciones:

E1

Causa: Existe un error con el script de detección de los sensores.

Posible solución: Detectar los errores y repararlos.

– RF9: Visualizar alertas de precauciones de cuidado de plantas

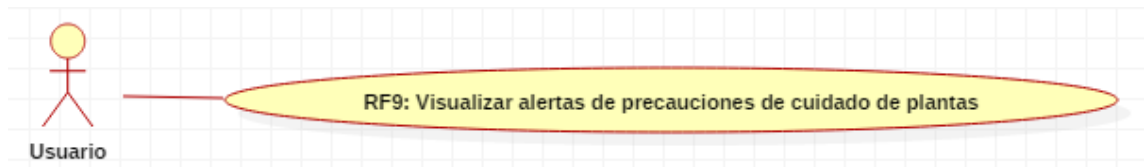


Figura 25: Diagrama visualizar alertas de precauciones de cuidado de plantas
Elaborado por: (Astudillo & Cadena, 2016)

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web.
2. El sistema genera la alerta de precaución, de acuerdo a los indicadores de cuidado de plantas. (E1)
3. El usuario visualiza un mensaje de alerta en la página web.

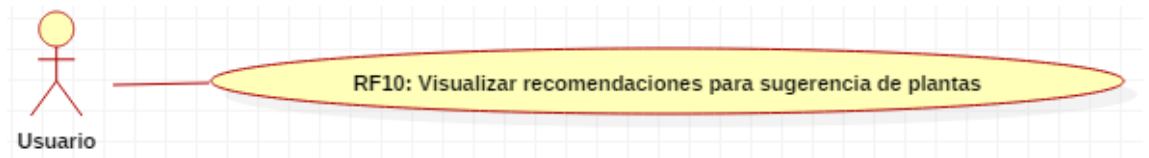
Excepciones:

E1

Causa: Existe un error con el script de detección de los sensores.

Posible solución: Detectar los errores y repararlos.

– RF10: Visualizar recomendaciones para sugerencia de plantas



*Figura 26: Visualizar recomendaciones para sugerencia de plantas
Elaborado por: (Astudillo & Cadena, 2016)*

Actor: Usuario

Flujo principal:

1. El usuario ingresa a la página web.
2. El usuario selecciona la opción Recomendaciones del menú principal para la sugerencia de plantas.
3. El sistema presenta preguntas generales para la sugerencia de plantas.
4. El usuario responde las preguntas de acuerdo a sus necesidades.
5. El usuario presiona el botón Recomendar.
6. El sistema presenta la sugerencia de plantas para el usuario. (E1)
7. El usuario visualiza en la página web las recomendaciones para el cuidado de las plantas.

Excepciones:

E1

Causa: Existe un error con la base de datos.

Posible solución: Detectar los errores y repararlos.

5.2: DISEÑO

Con el fin de especificar la estructura y el comportamiento de la integración de la página web y Arduino, se presenta el diagrama de arquitectura del mismo, con los componentes necesarios para su implementación.

5.2.1: DIAGRAMA DE ARQUITECTURA

El diagrama de arquitectura de la tercera iteración tiene como objetivo brindar una visión simplificada de lo que se desea construir para la integración de la página web y Arduino, el mismo que se presenta a continuación.

5.2.1.1: DIAGRAMA A DETALLE TERCERA ITERACIÓN

La tercera iteración al tratarse de una integración, es necesario incluir todos los componentes que permitirán el funcionamiento completo del prototipo. Como se puede apreciar en la Figura 27, en esta iteración actuarán los sensores de humedad y temperatura ambiental, de luz, humedad del suelo y movimiento, para la obtención de los datos por medio de Arduino, los cuales serán procesados por medio de un archivo php y presentados posteriormente en un archivo html. Además se contará con una base de datos que será gestionada para la visualización de datos almacenados. Finalmente el usuario será quien interactúe con la página web y sus funcionalidades respectivas.

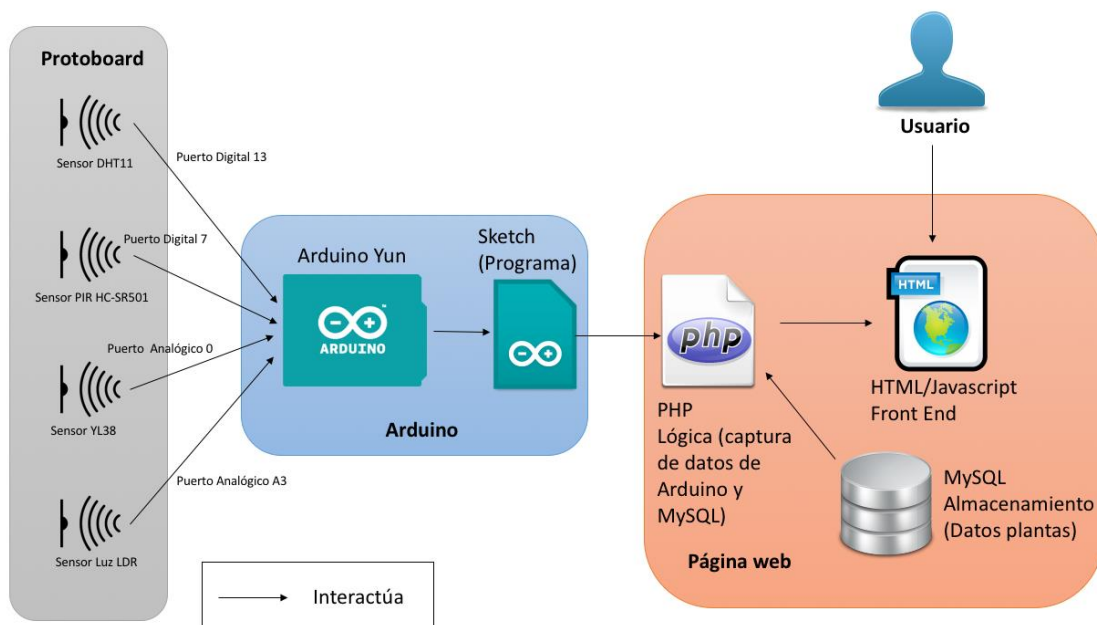


Figura 27: Diagrama a detalle tercera iteración
Elaborado por: (Astudillo & Cadena, 2016)

5.3: CODIFICACIÓN

5.3.1: Codificación Arduino Yún

Dentro de esta nueva iteración en relación al sketch (Programa incluido dentro de la placa Arduino), se modificó los datos presentados al usuario para poder facilitar la inclusión de estos datos dentro de la página web y permitir que las notificaciones puedan ser presentadas con datos de utilidad para los usuarios.

```
if (command == "temperatura") {  
    temperatura = dht.readTemperature();  
    client.println(temperatura);  
}  
  
if(command == "humedad"){  
    humedad = dht.readHumidity();  
    client.print(humedad);  
}  
  
if(command == "suelo"){  
    suelo = analogRead(A0);  
    if(suelo < 1000 && suelo >=600){  
        client.println("Seco");  
    }  
    if(suelo < 600 && suelo >=370){  
        client.println("Húmedo");  
    }  
    if(suelo < 370){  
        client.println("Excesiva");  
    }  
}
```

5.3.2: Codificación página web

Para lograr las funcionalidades de la página web, además del uso de HTML, CSS y PHP, se opta por el lenguaje de programación JavaScript.

5.3.2.1: *AlertaIndicadores.js*

Por medio de JavaScript, se consigue que los indicadores de cuidado de plantas se actualicen cada cierto tiempo a través de una función que reciba como parámetros el id del elemento HTML donde se mostrará el dato del indicador, los segundos en los que se actualizará el dato, y el archivo donde se encuentra almacenado el dato del indicador.

```
function refrescar(id,seg,url){  
    var id,seg,url,fetch_unix_timestamp;  
    var xmlhttp;  
    try{  
        xmlhttp=new XMLHttpRequest();  
    }  
    catch (e){ }  
    fetch_unix_timestamp = function(){  
        return parseInt(new Date().getTime().toString().substring(0, 10))  
    }  
    var nocacheurl = url+"?t="+fetch_unix_timestamp;  
    xmlhttp.onreadystatechange=function(){  
        if(xmlhttp.readyState == 4 && xmlhttp.status == 200){  
            document.getElementById(id).innerHTML=xmlhttp.responseText;  
            setTimeout(function(){refrescar(id,seg,url);},seg*1000);  
        }  
    }  
    xmlhttp.open("GET",nocacheurl,true);  
    xmlhttp.send(null);  
}
```

Además es posible la visualización de los mensajes de precaución y la alerta de acuerdo a las condiciones dadas para cada indicador, ya sea para temperatura, humedad, luz y humedad del suelo, a través de la implementación de una función que es llamada cada cierto tiempo y en donde se definen las limitantes para cada indicador.

Indicador humedad del suelo:

```
function alerta(){
    setTimeout(function(){ alerta(); }, 2500);
    var suelo = document.getElementById('humedadSuelo').textContent.trim();
    var indicadorSuelo = document.getElementById('indicador4').textContent;
    var imprimirAlerta = false;
    if(suelo == "Seco" || suelo == "Excesiva"){
        if(indicadorTemperatura != "Precaución"){
            imprimirAlerta = true;
            alertaMostrada = true;
            document.getElementById('indicador4').innerHTML = 'Precaución';
        }
    } else if(suelo != "Seco" || suelo != "Excesiva"){
        document.getElementById('indicador4').innerHTML = "";
        alertaSuelo = true;
    }
}
```

Con el fin de evitar que la alerta de precaución se genere continuamente para cada indicador, dentro de la función “alerta” se implementa una condición que indique que la alerta tan sólo aparezca una vez que uno de los indicadores active su mensaje de precaución.

```
if(imprimirAlerta){  
    if(alertaTemperatura && alertaHumedad && alertaLuz && alertaSuelo){  
        sweetAlert('Cuidado!','Por favor, revisa los indicadores');  
        alertaTemperatura = false;  
        alertaHumedad = false;  
        alertaLuz = false;  
        alertaSuelo = false;  
    }  
    imprimirAlerta = false;  
}
```

El archivo .js es importado en cada una de las páginas php creadas anteriormente, debido a que todas incluyen los datos procesados para la visualización de los indicadores de cuidado de plantas.

```
<script type="text/javascript" src="alertaIndicadores.js"></script>
```

5.3.2.2: *RecomendacionFinal.php*

De acuerdo a las respuestas seleccionadas por el usuario, y después de haber presionado en el botón “Recomendar”, se realiza una consulta a la base de datos donde se encuentran almacenadas diferentes tipos de plantas, junto con sus indicadores.


```
<?php

include "conexion.php";

$pregunta1= $_POST['radioPreguntaUno'];
$pregunta2= $_POST['radioPreguntaDos'];
$pregunta3= $_POST['radioPreguntaTres'];
$pregunta4= $_POST['radioPreguntaCuatro'];

if($pregunta4==19)
{
    $queryRecomendacion = "select NOMBREPLANTA,
INFOEXTRAPLANTA FROM PLANTA WHERE TIPOPLANTA = ".$pregunta1." AND
LUZPLANTA = ".$pregunta2." AND RIEGOSEMANALPLANTA = ".$pregunta3." AND
TEMPERATURAMAXPLANTA >= ".$pregunta4."" ;

    $resultadoRecomendacion = mysqli_query($conexion,$queryRecomendacion) or
die(mysqli_error($conexion));

} else {
    $queryRecomendacion = "select NOMBREPLANTA,
INFOEXTRAPLANTA FROM PLANTA WHERE TIPOPLANTA = ".$pregunta1." AND
LUZPLANTA = ".$pregunta2." AND RIEGOSEMANALPLANTA = ".$pregunta3." AND
TEMPERATURAMAXPLANTA <= ".$pregunta4."" ;

    $resultadoRecomendacion = mysqli_query($conexion,$queryRecomendacion) or
die(mysqli_error($conexion));

}

?>
```

El resultado de la consulta dependerá de las respuestas, y la sugerencia será mostrada en una tabla que contendrá el nombre de la planta e información extra de la misma.

```

<center><Table BORDER="1" cellspacing="3">
    <TR>
        <TH>NOMBRE PLANTA</TH>
        <TH>INFORMACIÓN ADICIONAL</TH>
    </TR>
    <?php
        while($auxPlantas = mysqli_fetch_array($resultadoRecomendacion))
        { ?>
            <TR>
                <TD><?PHP echo $auxPlantas['NOMBREPLANTA'];?></TD>
                <TD><?PHP echo $auxPlantas['INFOEXTRAPLANTA'];?></TD>
            </TR>
            <?php } ?>
    </Table></center>

```

5.4: PRUEBAS

5.4.1: Indicadores de cuidados de plantas

Acción	Ingresa a la página web, dirigiéndose a la opción Inicio del menú principal y visualizar los datos procesados de los indicadores de cuidados de plantas como: temperatura y humedad ambiental, luz ambiental y humedad del suelo.
Resultado Esperado	Se puede visualizar los datos procesados para los indicadores de cuidado de plantas de acuerdo al criterio dado para cada uno desde el navegador de un dispositivo.
Evidencia	

	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 14: Prueba indicadores de cuidados de plantas
Elaborado por: (Astudillo & Cadena, 2016)

5.4.2: Alertas de precauciones de cuidados de plantas

5.4.2.1: Mensajes de alerta

Acción	Ingresar a la página web y visualizar los mensajes de alerta de cuidados de plantas de acuerdo al estado de los indicadores.
Resultado Esperado	Se puede visualizar el texto previsto para los mensajes de alerta de cuidado de plantas de acuerdo a las limitantes de cada indicador desde el navegador de un dispositivo.
Evidencia	

	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 15: Prueba mensajes de alerta
Elaborado por: (Astudillo & Cadena, 2016)

5.4.2.2: Alerta

Acción	Ingresar a la página web y visualizar la alerta de cuidados de plantas, una vez que uno de los indicadores haya presentado su mensaje de “Precaución”.
Resultado Esperado	Se puede visualizar la alerta de cuidado de plantas desde el navegador de un dispositivo.
Evidencia	

	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 16: Prueba alerta
Elaborado por: (Astudillo & Cadena, 2016)

5.4.3: Recomendaciones para sugerencia de plantas

Acción	Ingresar a la página web, dirigirse a la opción Recomendaciones del menú principal, visualizar el cuestionario y responder las preguntas propuestas. Presionar en el botón “Recomendar” y visualizar la sugerencia de plantas.
Resultado Esperado	Se puede visualizar el cuestionario propuesto y una tabla con la sugerencia de plantas desde el navegador de un dispositivo.
Evidencia	

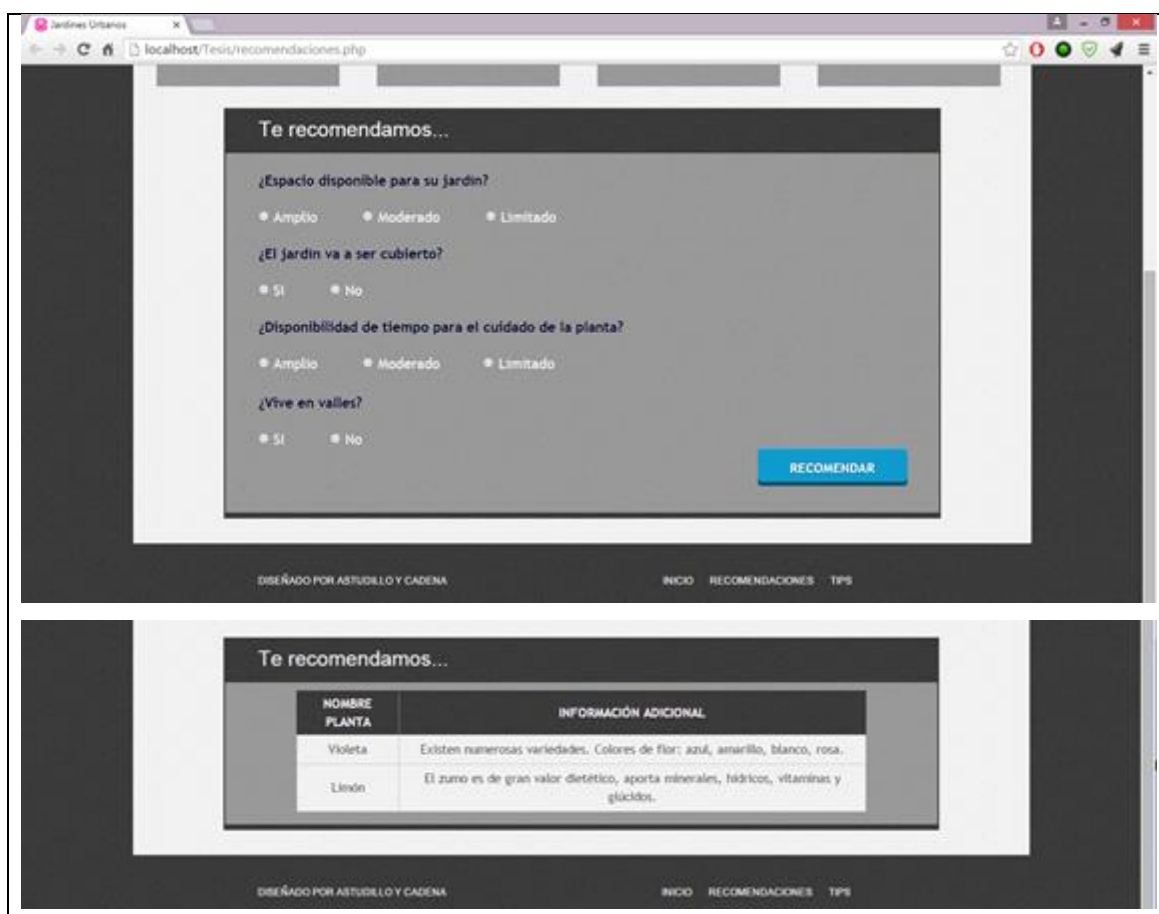
 <p>The top screenshot shows a web browser window with the URL 'localhost/Tesis/recomendaciones.php'. The page has a dark header with the title 'Jardines Urbanos'. The main content area is titled 'Te recomendamos...' and contains four questions with radio button options:</p> <ul style="list-style-type: none"> ¿Espacio disponible para su jardín? (Amplio, Moderado, Limitado) ¿El jardín va a ser cubierto? (Sí, No) ¿Disponibilidad de tiempo para el cuidado de la planta? (Amplio, Moderado, Limitado) ¿Vive en valles? (Sí, No) <p>A blue 'RECOMENDAR' button is at the bottom right. The footer contains 'DISEÑADO POR ASTUDILLO Y CADENA' and navigation links 'INICIO', 'RECOMENDACIONES', and 'TIPS'.</p> <p>The bottom screenshot shows the result of the recommendation. It features a table with two columns: 'NOMBRE PLANTA' and 'INFORMACIÓN ADICIONAL'.</p> <table border="1"> <thead> <tr> <th>NOMBRE PLANTA</th> <th>INFORMACIÓN ADICIONAL</th> </tr> </thead> <tbody> <tr> <td>Violeta</td> <td>Existen numerosas variedades. Colores de flor: azul, amarillo, blanco, rosa.</td> </tr> <tr> <td>Lirio</td> <td>El zumo es de gran valor dietético, aporta minerales, hidratos, vitaminas y glúcidos.</td> </tr> </tbody> </table> <p>The footer is identical to the first screenshot.</p>	NOMBRE PLANTA	INFORMACIÓN ADICIONAL	Violeta	Existen numerosas variedades. Colores de flor: azul, amarillo, blanco, rosa.	Lirio	El zumo es de gran valor dietético, aporta minerales, hidratos, vitaminas y glúcidos.	<p>Resultado Se pasó la prueba satisfactoriamente</p>
NOMBRE PLANTA	INFORMACIÓN ADICIONAL						
Violeta	Existen numerosas variedades. Colores de flor: azul, amarillo, blanco, rosa.						
Lirio	El zumo es de gran valor dietético, aporta minerales, hidratos, vitaminas y glúcidos.						

Tabla 17: Prueba recomendaciones para sugerencia de plantas
Elaborado por: (Astudillo & Cadena, 2016)

5.4.4: Sensor de humedad y temperatura

Acción	Consultar la temperatura en la ruta <i>IPArduino/arduino/temperatura/13</i> , cuando el sensor DTH11 esté conectado al Arduino con el sketch que contiene la función para leer la temperatura
Resultado Esperado	Se puede visualizar los cambios del dato temperatura ambiental desde el navegador de un dispositivo
Evidencia	

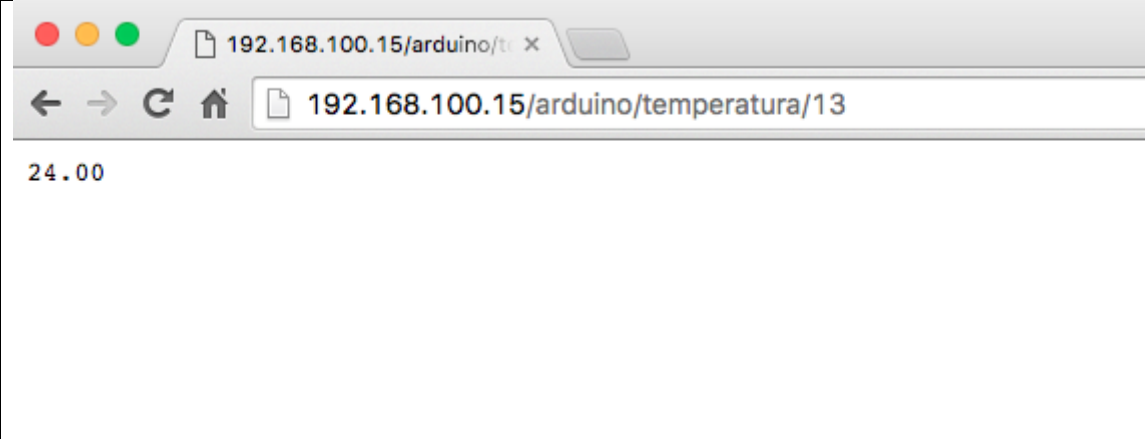
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 18: Tercera prueba sensor de temperatura
Elaborado por: (Astudillo & Cadena, 2016)

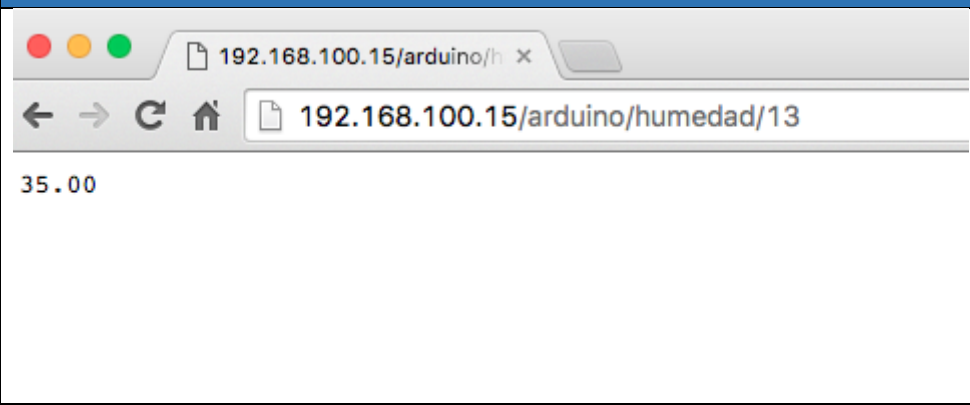
Acción	Consultar la humedad en la ruta <i>IPArduino/arduino/humedad/13</i> , cuando el sensor DTH11 esté conectado al Arduino con el sketch que contiene la función para leer la humedad
Resultado Esperado	Se puede visualizar los cambios del dato humedad ambiental desde el navegador de un dispositivo
Evidencia	
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 19: Tercera prueba sensor de humedad
Elaborado por: (Astudillo & Cadena, 2016)

5.4.5: Sensor de humedad del suelo


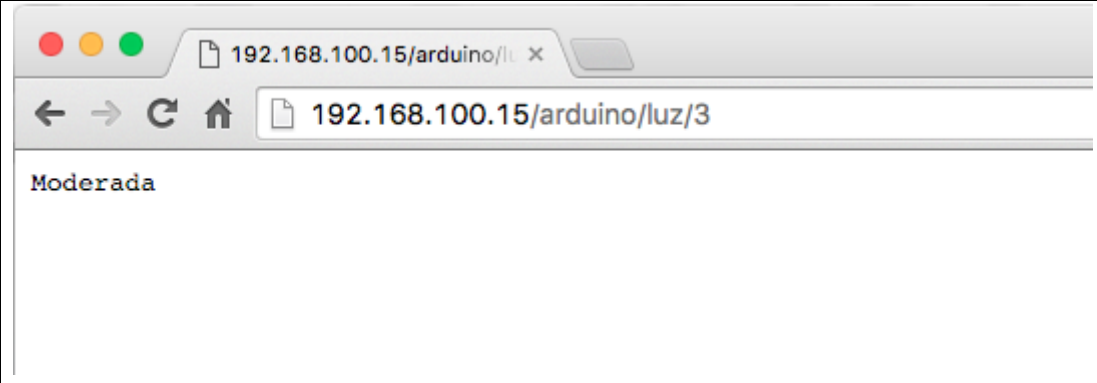
Acción	Consultar la humedad del suelo en la ruta <i>IPArduino/arduino/suelo/0</i> , cuando el sensor DTH11 esté conectado al Arduino con el sketch que contiene la función para leer la humedad del suelo
Resultado Esperado	Se puede visualizar los cambios de la humedad del suelo desde el navegador de un dispositivo
Evidencia	
	
Resultado	Se pasó la prueba satisfactoriamente

Tabla 20: Tercera prueba sensor de humedad del suelo
Elaborado por: (Astudillo & Cadena, 2016)

5.4.6: Sensor de Luz

Acción	Consultar la luz en la ruta <i>IPArduino/arduino/luz/3</i> , cuando el sensor DTH11 esté conectado al Arduino con el sketch que contiene la función para leer la luz
Resultado Esperado	Se puede visualizar la luz ambiental desde el navegador de un dispositivo
Evidencia	

	
Resultado	Se pasó la prueba satisfactoriamente

*Tabla 21: Tercera prueba sensor de luz
Elaborado por: (Astudillo & Cadena, 2016)*

CONCLUSIONES

- Las herramientas libres disponibles, permiten una gran variedad de soluciones a distintos problemas, como es el caso de la automatización mediante sensores. La comunidad de software y hardware libre permite que el aprendizaje de una nueva herramienta pueda ser en un menor tiempo.
- La información disponible para herramientas con licencia de software libre tiene una mayor accesibilidad y en mayor cantidad, gracias a foros y blogs que comparten experiencias y soluciones a problemas.
- Las herramientas libres, al facilitar la modificación de funcionalidades disponibles, permite que la herramienta sea flexible, adaptándose para poder cumplir con las necesidades y requerimientos de un determinado problema.
- Debido a la escasa demanda de componentes electrónicos existente en el mercado de Ecuador, en el caso de sensores, no se tiene gran variedad de productos al momento de adquirirlos, lo cual restringe la solución a utilizarse.
- La integración existente entre las tecnologías actuales orientadas a servidor-cliente, permiten una gran flexibilidad al momento de implementar funcionalidades que requiera el sistema a desarrollarse.
- El uso de una metodología incremental, permitió una mayor flexibilidad en cuanto a cambios dentro de los requerimientos del desarrollo, ya que al no tenerlos concretamente definidos al inicio, existió la posibilidad de ir adaptándolos a las funcionalidades del sistema.
- La escasa información de los sensores compatibles con la plataforma Arduino, se convierte en una restricción al momento de querer aprovechar al máximo las capacidades de los mismos.
- La creación de lenguajes de programación exclusivos para herramientas de desarrollo, puede llegar a ser un limitante durante la implementación, debido a sus características propias, a pesar de compartir paradigmas.

RECOMENDACIONES

- Con la finalidad de contar con una conexión óptima de los sensores, se recomienda la construcción de una placa de circuitos impresa para evitar complicaciones entre los cables necesarios conectados a los sensores, y así mejorar la organización y fijación de los mismos.
- Para permitir que el sistema sea capaz de monitorear mayor cantidad de jardines, es necesario modificar la visualización actual en la página web, dependiendo del total de espacios a controlar.
- Para mejorar las notificaciones existentes para el cuidado de las plantas, se recomienda que las alertas permitan generar recordatorios, para que el usuario pueda tener un mayor seguimiento y control de sus jardines.
- Para poder ampliar el alcance del prototipo, se puede implementar una funcionalidad adicional que permita el uso de nuevas tecnologías, como es el caso de dispositivos móviles, que facilite la lectura de recordatorios para cuidado de los jardines.
- Para que el usuario pueda tener mayor precaución con cualquier tipo de amenaza cercana a sus jardines, se propone incluir una sirena que funcione de acuerdo al sensor de movimiento.
- Para complementar la experiencia del usuario con el sistema, se propone realizar una implementación de una versión dirigida a smartphones, con el fin de que el usuario pueda efectuar un seguimiento de sus jardines de una manera más sencilla.
- Para poder expandir las funcionalidades del sistema, se puede incluir la opción de manejar usuarios, para que éstos tengan la posibilidad de mantener privada la información de sus jardines y tengan un acceso exclusivo.
- Considerando las recomendaciones anteriores, como futuras versiones del sistema se puede pensar en la opción de alojarlo en un hosting con su propio dominio, con el objetivo de que no sólo esté disponible en una red local.

BIBLIOGRAFÍA

- (s.f.). Obtenido de Arduino: <http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure>
- (s.f.). Obtenido de Nextia Fenix: <http://www.nextiafenix.com/producto/sensor-pir-de-movimiento-hc-sr501/>
- (s.f.). Obtenido de Electrónicos Caldas: <http://www.electronicoscaldas.com/sensores-de-movimiento/415-sensor-de-movimiento-piroelectrico-infrarrojo-hc-sr501.html>
- (s.f.). Obtenido de Instructables: <http://www.instructables.com/id/Ejemplo-b%C3%A1sico-con-HC-SR501/>
- (s.f.). Obtenido de Electronilab: <http://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/>
- (s.f.). Obtenido de Naylamp Mechatronics: <http://www.naylampmechatronics.com/luz-y-sonido/76-modulo-sensor-de-luz-digital-bh1750.html>
- (s.f.). Obtenido de Patagoniatec: <http://saber.patagoniatecnology.com/sensor-digital-de-luz-bh1750/>
- (s.f.). Obtenido de Patagoniatec: <http://saber.patagoniatecnology.com/sensor-digital-de-luz-bh1750/>
- (s.f.). Obtenido de Educatrónica: http://educatronica.blogspot.com/p/blog-page_29.html
- (s.f.). Obtenido de Talos Electronics: <https://www.taloselectronics.com/producto/sensor-de-humedad-del-suelo/>
- (s.f.). Obtenido de Infojardín: <http://articulos.infojardin.com/plantas/plantas.htm>
- (s.f.). Obtenido de Ingeniatic: <http://unicrom.com/ldr-fotorresistencia-fotorresistor/>
- (s.f.). Obtenido de Como hacer un robot: <http://www.comohacerturobot.com/Taller/taller-sensorluminosidad.htm>
- (19 de Septiembre de 2014). Obtenido de GeekBot: <http://www.geekbotelectronics.com/dht11-sensor-de-humedad-y-temperatura/>

- (7 de Septiembre de 2015). Obtenido de Arduino:
<https://www.arduino.cc/en/Guide/Environment>
- (16 de Junio de 2015). Obtenido de Huerto urbano en mi balcón:
<http://www.huertourbanoenmibalcon.es/que-es-un-huerto-urbano/>
- Adafruit. (2016). *Raspberry PI*. Obtenido de Adafruit:
<https://www.adafruit.com/category/105>
- Agromática. (29 de Abril de 2014). *Humedad del suelo. Cómo se comporta y su importancia: Agromática*. Obtenido de <http://www.agromatica.es/humedad-del-suelo/>
- Almorox, J. (Abril de 2007). Obtenido de OCW Consortium:
<http://ocw.upm.es/ingenieria-agroforestal/climatologia-aplicada-a-la-ingenieria-y-medioambiente/contenidos/tema-4/TEMPERATURA-DEL-SUELO.pdf>
- Amangandi, J. (5 de Octubre de 2012). Obtenido de
<http://jamangandi2012.blogspot.com/2012/10/que-es-arduino-te-lo-mostramos-en-un.html>
- Arduino. (s.f.). Obtenido de Arduino: <http://playground.arduino.cc/Interfacing/Java>
- Arduino: *Tecnología para todos*. (s.f.). Obtenido de
<http://arduinodehtics.weebly.com/historia.html>
- Boisset, C. (1995). *Jardín y arquitectura. Guía completa de planificación y plantación del jardín*. (U. Fischer, Trad.) Barcelona: BLUME.
- Buechel, T. (17 de Septiembre de 2015). Obtenido de PROMIX:
<http://www.pthorticulture.com/es/centro-de-formacion/consejos-para-la-propagacion-exitosa-de-plantas-jovenes/>
- Cabezas Granado, L. M., & González Lozano, F. J. (2015). *Desarrollo Web con PHP Y MySQL*. Madrid, España: Grupo Anaya.
- CANNA. (2015). Obtenido de CANNA:
http://www.canna.es/influencia_temperatura_ambiental_en_las_plantas
- Cantudo, C. (25 de Marzo de 2011). *Los 10 cuidados de un jardín: Soy Manitas*. Obtenido de <http://www.soymanitas.com/los-10-cuidados-de-un-jardin>

Carabal, G. (6 de Febrero de 2011). Obtenido de <http://electronicaradical.blogspot.com/2011/02/fotoresistencia-ldr.html>

Cekit. (s.f.). *Curso practico de electronica moderna* .

Couto, J. (24 de Diciembre de 2012). Obtenido de Taller Arduino: <http://tallerarduino.com/2012/12/24/sensor-dht11-humedad-y-temperatura-con-arduino/>

Díaz Díaz, J. (31 de Marzo de 2014). Massimo banzi, el hombre que ha puesto el open hardware al alcance de todos. *Cinco Dias*. Obtenido de <http://search.proquest.com/docview/1522427156?accountid=13357>

Ecoterrazas. (14 de Junio de 2013). *Cuidados de verano para tu jardín; Ecoterrazas*. Obtenido de <http://www.ecoterrazas.com/blog/cuidados-de-verano-para-tu-jardin/#comments>

Enríquez Herrador, R. (13 de Noviembre de 2009). Obtenido de Aula de Software Libre de la Universidad de Córdoba: http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf

García González, A. (14 de Febrero de 2013). Obtenido de Panama Hitek: <http://panamahitek.com/dht11-sensor-de-humedadtemperatura-para-arduino/>

Gilfillan, I. (s.f.). *La Biblia MySQL*. Anaya Multimedia.

Giraldo, S. (26 de Marzo de 2011). *Humedad en invernaderos: Invernaderos & jardines*. Obtenido de <http://www.invernaderosyjardines.com/component/k2/item/13-humedadinvernaderos.html>

Grajales-Hall, M. (23 de Mayo de 2002). Diez pasos simples para tener un hermoso jardín. *La Opinión*(250). Obtenido de <http://search.proquest.com/docview/368347052/abstract?accountid=13357>

Group, T. P. (s.f.). *Historia de PHP* . Recuperado el 15 de 11 de 2015, de PHP: <http://php.net/manual/es/history.php.php>

Historia de PHP: php. (2015). Obtenido de <http://php.net/manual/es/history.php.php>

Historia lenguajes de programación. (s.f.). Obtenido de <http://elzoo.es/archivos/Libros/PROGRAMACIÓN/Historia%20de%20los%20lenguajes%20de%20programación.pdf>

Historia PHP. (s.f.). Obtenido de <http://php.net/manual/es/history.php>

Ibáñez, J. J. (21 de Enero de 2008). *Temperatura del suelo y Microclimatología: Miod.* Obtenido de <http://www.madrimasd.org/blogs/universo/2008/01/21/82825>

Infojardín. (2015). *Diseño del jardín: clima y suelo: Infojardín.* Obtenido de <http://articulos.infojardin.com/articulos/disenio-jardin-clima-suelo.htm>

Ingeniería del software: Metodologías y ciclos de vida. (Marzo de 2009). pág. 30.

Introducing MySQL Connector/Arduino. (2013). Obtenido de http://drcharlesbell.blogspot.com/2013/04/introducing-mysql-connectorarduino_6.html

Letrán, J. (2011). Obtenido de Corto Circuito: <http://www.cortoc.com/2011/12/introduccion-arduino.html>

Liechti, C. (s.f.). *PySerial.* Obtenido de <https://pythonhosted.org/pyserial/>

Littlebits Electronics. (2016). *What is Littlebits?* Obtenido de Littlebits: <http://littlebits.cc/how-it-works>

López Medina, P. (2008). Parques y jardines urbanos. Espacios para la sostenibilidad. *Medio Ambiente.*

Lopez, A. (s.f.). *Introducción a Java.* Obtenido de <https://aureliux.files.wordpress.com/2008/07/curso1-3.pdf>

Monsalve, A. (23 de Enero de 2001). Negocios de Internet: Que es un PHP? *La Opinión.* Obtenido de <http://search.proquest.com/docview/368378096?accountid=13357>

Perez, N. (2011). Obtenido de Modelos de Proceso: <https://sistinfii.files.wordpress.com/2011/03/siii2011-02-modelos-de-proceso.pdf>

Procesos software. (s.f.). Obtenido de Modelo incremental: <https://procesossoftware.wikispaces.com/Modelo+Incremental>

Prometec. (s.f.). Obtenido de Prometec: <http://www.prometec.net/sensores-dht11/>

Revisión de las características hardware de la nueva Arduino Yún. (s.f.). Obtenido de Mexchip: <http://www.mexchip.com/2013/09/revision-de-las-caracteristicas-hardware-de-la-nueva-arduino-yun/>

Revuelta, K. (11 de Enero de 2011). Obtenido de Project Bot: <http://projectbot.blogspot.com/2011/01/sobre-arduino-y-su-lenguaje.html>

Reyna, T., Reyna, S., Lábaque, M., Fulginiti, F., Riha, C., & Linares, J. (2011). Obtenido de ambi-agua: <http://www.ambi-agua.net/seer/index.php/ambi-agua/article/download/496/864>.

Roger, P. (2005). *Ingeniería del software. Un enfoque práctico*. México D.F: McGraw - Hil.

Sánchez Onofre, J. (04 de Julio de 2014). Arduino: El corazón libre del mundo hiperconectado. *Economista*. Obtenido de <http://search.proquest.com/docview/1542600662?accountid=13357>

Schach, S. (2006). *Ingeniería de software clásica y orientada a objetos*. México D.F: McGraw - Hill.

Stephen, S. (2006). *Ingeniería de software clásica y orientada a objetos*. Mexico D.F: Mc Graw Hill.

Torrente Artero, Ó. (2013). *ARDUINO. Curso práctico de formación*. RC Libros. Obtenido de https://books.google.com.ec/books?id=6cZhDmf7suQC&pg=PA68&lpg=PA68&dq=Arduino+y+el+software&source=bl&ots=AZcEjXJCwI&sig=eCcnvLoVCpMLXD11zE8Nq3p7HAU&hl=en&sa=X&ved=0CF0Q6AEwDGoVChMIv_n0uZ-LyQIVQTomCh2H5AOh#v=onepage&q&f=false

Torres, M. (2007). *Tutorial microcontroladores PIC*. Obtenido de http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf

Welling, L., & Thomson, L. (2005). *Desarrollo Web con PHP y MySQL*. Madrid, España: Grupo Anaya.

Wilde, K. (2008). *Mi primer jardín. Guía práctica.* (R. Martínez Castellote, Trad.)
Barcelona: BLUME.

ANEXOS

ANEXO 1: Código fuente sketch Arduino Yún primera iteración

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <DHT.h>
#define DHTPIN 13
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

YunServer server;
float temperatura = 0;
float humedad = 0;
int suelo = 0;
int luz = 0;

void setup() {
  Serial.begin(9600);
  pinMode(13, INPUT); //sensor humedad y temperatura
  pinMode(A0, INPUT); // sensor de suelo
  pinMode(A3, INPUT); //sensor de luz
  pinMode(7, INPUT); //sensor movimiento
  pinMode(11, OUTPUT); //led
  Bridge.begin();
  server.listenOnLocalhost();
  server.begin();
  dht.begin();
}

void loop() {
  YunClient client = server.accept();
  if (client) {
    process(client);
    client.stop();
  }
  if(digitalRead(7) == HIGH)
  {
    digitalWrite(11,HIGH);
  }
}
```

```
        delay(1000);
        digitalWrite(11,LOW);
    }
    delay(50);
}

void process(YunClient client) {
    String command = client.readStringUntil('/');

    if (command == "temperatura") {
        temperatura = dht.readTemperature();
        client.print("Temperatura: ");
        client.println(temperatura);
    }

    if(command == "humedad"){
        humedad = dht.readHumidity();
        client.print("Humedad: ");
        client.print(humedad);
    }

    if(command == "suelo"){
        suelo = analogRead(A0);
        client.print("Humedad del suelo: ");
        client.println(suelo);
    }

    if(command == "luz"){
        luz = analogRead(A3);
        client.print("luz: ");
        client.println(luz);
    }
}
```

ANEXO 2: Código fuente página web segunda iteración

Inicio.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />

</head>
<body>
<div id="container">
  <div id="containertop"></div>
  <h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
  <hr />
  <div id="navigation_wrapper">
    <ul id="navigation">
      <li><a href="inicio.php" title="Inicio">Inicio</a></li>
      <li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
      <li><a href="tips.php" title="Tips">Tips</a></li>
      <li><a href="" title="Acerca de">Acerca de</a></li>
    </ul>
  </div>

  <div id="banner_wrapper">
    <div class="banner">
      <h2>Temperatura <br /> Ambiental</h2>
      <p>&nbsp;</p>
      <div class="image_wrapper">
        <p></p>
        <p>&nbsp;</p>
        <form id="form2" name="form2" method="post" action="">
          <p>
            <label for="temperaturaAmbiental"></label>
            <input type="text" class="textbox" name="temperaturaAmbiental"
id="temperaturaAmbiental" readonly="readonly" />
          </p>
          <p>&nbsp;</p>
          <label class="alerta" id="indicador1"></label>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

```
</form>
</div>
</div>
<div class="banner">
  <h2>Humedad <br />Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form3" name="form3" method="post" action="">
      <p>
        <label for="humedadAmbiental"></label>
        <input type="text" class="textbox" name="humedadAmbiental"
id="humedadAmbiental" readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador2"></label>
    </form>
  </div>
</div>
<div class="banner">
  <h2>Luz <br />Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form1" name="form1" method="post" action="">
      <p>
        <label for="luzAmbiental"></label>
        <input name="luzAmbiental" type="text" class="textbox" id="luzAmbiental"
readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador3"></label>
    </form>
  </div>
</div>
<div class="banner banner_last">
  <h2>Humedad <br /> Suelo</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form4" name="form4" method="post" action="">
      <p>
        <label for="humedadSuelo"></label>
        <input type="text" class="textbox" name="humedadSuelo" id="humedadSuelo"
readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador4"></label>
    </form>
  </div>
</div>
```

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<hr />
</div>
</div>

<div id="footer_wrapper">
<div id="footer">
<div class="logo">Disẽado por astudillo y cadena</div>
<ul>
<li>inicio</li>
<li>recomendaciones</li>
<li>tips</li>
<li>acerca de</li>
</ul>
</div>
</div></body>
</html>
```

Recomendaciones.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="sweetalert/dist/sweetalert.css">
<script src="sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript">
function checkCookies() {
sweetAlert('Cuidado!', 'Por favor, revisa tus indicadores');}
window.onload = checkCookies;
</script>

</head>

<body onLoad="checkCookies()">
<div id="container">
<div id="containertop"></div>
<h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
<hr />
<div id="navigation_wrapper">
<ul id="navigation">
<li><a href="inicio.php" title="Inicio">Inicio</a></li>
<li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
<li><a href="tips.php" title="Tips">Tips</a></li>
<li><a href="" title="Acerca de">Acerca de</a></li>
</ul>
</div>
```

```
<div id="banner_wrapper">
  <div class="banner">
    <h2>Temperatura <br /> Ambiental</h2>
    <p>&nbsp;</p>
    <div class="image_wrapper">
      <p></p>
      <p>&nbsp;</p>
      <form id="form2" name="form2" method="post" action="">
        <p>
          <label for="temperaturaAmbiental"></label>
          <input type="text" class="textbox" name="temperaturaAmbiental"
id="temperaturaAmbiental" readonly="readonly" />
        </p>
        <p>&nbsp;</p>
        <label class="alerta" id="indicador1"></label>
      </form>
    </div>
  </div>
  <div class="banner">
    <h2>Humedad <br /> Ambiental</h2>
    <p>&nbsp;</p>
    <div class="image_wrapper">
      <p></p>
      <p>&nbsp;</p>
      <form id="form3" name="form3" method="post" action="">
        <p>
          <label for="humedadAmbiental"></label>
          <input type="text" class="textbox" name="humedadAmbiental"
id="humedadAmbiental" readonly="readonly" />
        </p>
        <p>&nbsp;</p>
        <label class="alerta" id="indicador2"></label>
      </form>
    </div>
  </div>
  <div class="banner">
    <h2>Luz <br /> Ambiental</h2>
    <p>&nbsp;</p>
    <div class="image_wrapper">
      <p></p>
      <p>&nbsp;</p>
      <form id="form1" name="form1" method="post" action="">
        <p>
          <label for="luzAmbiental"></label>
          <input name="luzAmbiental" type="text" class="textbox" id="luzAmbiental"
readonly="readonly" />
        </p>
        <p>&nbsp;</p>
        <label class="alerta" id="indicador3"></label>
      </form>
    </div>
  </div>
  <div class="banner banner_last">
    <h2>Humedad <br /> Suelo</h2>
```

```
<p>&nbsp;</p>
<div class="image_wrapper">
  <p></p>
  <p>&nbsp;</p>
  <form id="form4" name="form4" method="post" action="">
    <p>
      <label for="humedadSuelo"></label>
      <input type="text" class="textbox" name="humedadSuelo" id="humedadSuelo"
readonly="readonly" />
    </p>
    <p>&nbsp;</p>
    <label class="alerta" id="indicador4"></label>
  </form>
</div>
</div>

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<hr />
</div>

<div id="services">
  <h2>Te recomendamos...<span></span></h2>
  <div id="servicescontent">
    <form class="form" id="form1" name="form1" method="post"
action="recomendacionFinal.php">
      <p>
        <label class="textPregunta" >¿Espacio disponible para su jardín?</label>
      </p>
      <p>&nbsp;</p>
      <p>
        <label class="radioPregunta">
          <input type="radio" name="radioPreguntaUno" value="árbol"
id="radioPreguntaUno_0" />
          Amplio </label>

          <label class="radioPregunta">
            <input type="radio" name="radioPreguntaUno" value="arbusto"
id="radioPreguntaDos_0" />
            Moderado </label>

            <label class="radioPregunta">
              <input type="radio" name="radioPreguntaUno" value="herbácea"
id="radioPreguntaTres_1" />
              Limitado</label>
            </p>
            <p>&nbsp;</p>
            <p>
              <label class="textPregunta" >¿El jardín va a ser cubierto?</label>
            </p>
            <p>&nbsp;</p>
            <p>
              <label class="radioPregunta">
```

```

        <input type="radio" name="radioPreguntaDos" value="moderada"
id="radioPreguntaCuatro_0" />
        Si </label>
        <label class="radioPregunta">
        <input type="radio" name="radioPreguntaDos" value="abundante"
id="radioPreguntaCinco_1" />
        No</label>
    </p>
    <p>&nbsp;</p>

    <p>
        <label class="textPregunta" >¿Disponibilidad de tiempo para el cuidado de la
planta?</label>
    </p>
    <p>&nbsp;</p>
    <p>
        <label class="radioPregunta">
        <input type="radio" name="radioPreguntaTres" value="3" id="radioPreguntaSeis_0" />
        Amplio </label>
        <label class="radioPregunta">
        <input type="radio" name="radioPreguntaTres" value="2" id="radioPreguntaSiete_1" />
        Moderado</label>
        <label class="radioPregunta">
        <input type="radio" name="radioPreguntaTres" value="1" id="radioPreguntaOcho_1" />
        Limitado</label>
    </p>
    <p>&nbsp;</p>
    <p>
        <label class="textPregunta" >¿Vive en valles?</label>
    </p>
    <p>&nbsp;</p>
    <p>
        <label class="radioPregunta">
        <input type="radio" name="radioPreguntaCuatro" value="19"
id="radioPreguntaNueve_0" />
        Si </label>
        <label class="radioPregunta">
        <input type="radio" name="radioPreguntaCuatro" value="18"
id="radioPreguntaDiez_1" />
        No</label>
    </p>

    <p>
        <input type="submit" class="button" name="botonSubmitRecomendaciones"
id="botonSubmitRecomendaciones" value="Recomendar" />
    </p>
    <p> <br />
    </p>
</form>
</div>
<span class="bottom_rounded"></span>
<hr />
</div>
</div>

```



```
<div id="footer_wrapper">
  <div id="footer">
    <div class="logo">Disenado por astudillo y cadena</div>
    <ul>
      <li>inicio</li>
      <li>recomendaciones</li>
      <li>tips</li>
      <li>acerca de</li>
    </ul>
  </div>
</div></body>
</html>
```

Tips.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="container">
  <div id="containertop"></div>
  <h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
  <hr />
  <div id="navigation_wrapper">
    <ul id="navigation">
      <li><a href="inicio.php" title="Inicio">Inicio</a></li>
      <li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
      <li><a href="tips.php" title="Tips">Tips</a></li>
      <li><a href="" title="Acerca de">Acerca de</a></li>
    </ul>
  </div>

  <div id="banner_wrapper">
    <div class="banner">
      <h2>Temperatura <br /> Ambiental</h2>
      <p>&nbsp;</p>
      <div class="image_wrapper">
        <p></p>
        <p>&nbsp;</p>
        <form id="form2" name="form2" method="post" action="">
          <p>
            <label for="temperaturaAmbiental"></label>
            <input type="text" class="textbox" name="temperaturaAmbiental"
id="temperaturaAmbiental" readonly="readonly" />
          </p>
          <p>&nbsp;</p>
          <label class="alerta" id="indicador1"></label>
```

```

    </form>
  </div>
</div>
<div class="banner">
  <h2>Humedad <br />Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form3" name="form3" method="post" action="">
      <p>
        <label for="humedadAmbiental"></label>
        <input type="text" class="textbox" name="humedadAmbiental"
id="humedadAmbiental" readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador2"></label>
    </form>
  </div>
</div>
<div class="banner">
  <h2>Luz <br />Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form1" name="form1" method="post" action="">
      <p>
        <label for="luzAmbiental"></label>
        <input name="luzAmbiental" type="text" class="textbox" id="luzAmbiental"
readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador3"></label>
    </form>
  </div>
</div>
<div class="banner banner_last">
  <h2>Humedad <br /> Suelo</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form4" name="form4" method="post" action="">
      <p>
        <label for="humedadSuelo"></label>
        <input type="text" class="textbox" name="humedadSuelo" id="humedadSuelo"
readonly="readonly" />
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador4"></label>
    </form>
  </div>
</div>

```

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p></p>
<p>&nbsp;</p>
<hr />
</div>
</div>

<div id="footer_wrapper">
  <div id="footer">
    <div class="logo">Dise&ntilde;ado por astudillo y cadena</div>
    <ul>
      <li>inicio</li>
      <li>recomendaciones</li>
      <li>tips</li>
      <li>acerca de</li>
    </ul>
  </div>
</div></body>
</html>
```

ANEXO 3: Código fuente sketch Arduino Yún tercera iteración

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <DHT.h>
#define DHTPIN 13
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

YunServer server;
float temperatura = 0;
float humedad = 0;
int suelo = 0;
int luz = 0;

void setup() {
  Serial.begin(9600);
  pinMode(13, INPUT); //sensor humedad y temperatura
  pinMode(A0, INPUT); // sensor de suelo
  pinMode(A3, INPUT); //sensor de luz
  pinMode(7, INPUT); //sensor movimiento
  pinMode(11, OUTPUT); //led
  Bridge.begin();
  server.listenOnLocalhost();
  server.begin();
  dht.begin();
}

void loop() {
  YunClient client = server.accept();
  if (client) {
    process(client);
    client.stop();
  }
  if(digitalRead(7) == HIGH)
  {
    digitalWrite(11,HIGH);
    delay(1000);
    digitalWrite(11,LOW);
  }
}
```

```
delay(50);
}

void process(YunClient client) {
  String command = client.readStringUntil('/');

  if (command == "temperatura") {
    temperatura = dht.readTemperature();
    client.println(temperatura);
  }

  if(command == "humedad"){
    humedad = dht.readHumidity();
    client.print(humedad);
  }

  if(command == "suelo"){
    suelo = analogRead(A0);
    if(suelo < 1000 && suelo >=600){
      client.println("Seco");
    }
    if(suelo < 600 && suelo >=370){
      client.println("Húmedo");
    }
    if(suelo < 370){
      client.println("Excesiva");
    }
  }

  if(command == "luz"){
    luz = analogRead(A3);
    if(luz < 1000 && luz >=500){
      client.println("Abundante");
    }
    if(luz < 500 && luz >=250){
      client.println("Moderada");
    }
    if(luz < 250){
      client.println("Limitada");
    }
  }
}
```

ANEXO 4: Código fuente página web tercera iteración

Inicio.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="UTF-8" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="sweetalert/dist/sweetalert.css">
<script src="sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript" src="alertaIndicadores.js"></script>
</head>
<body>
<div id="container">
<div id="containertop"></div>
<h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
<hr />
<div id="navigation_wrapper">
<ul id="navigation">
<li><a href="inicio.php" title="Inicio">Inicio</a></li>
<li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
<li><a href="tips.php" title="Tips">Tips</a></li>
</ul>
</div>

<div id="banner_wrapper">
<div class="banner">
<h2>Temperatura <br /> Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form2" name="form2" method="post" action="">
<p>
<label for="temperaturaAmbiental"></label>
<center><label class="textbox" id="temperaturaAmbiental"></label></center>
</p>
</div>
</div>
</div>
```

```

        <p>&nbsp;</p>
        <label class="alerta" id="indicador1"></label>
    </form>
</div>
</div>
<div class="banner">
    <h2>Humedad <br />Ambiental</h2>
    <p>&nbsp;</p>
    <div class="image_wrapper">
        <p></p>
        <p>&nbsp;</p>
        <form id="form3" name="form3" method="post" action="">
            <p>
                <label for="humedadAmbiental"></label>
                <center><label class="textbox" id="humedadAmbiental"></label></center>
            </p>
            <p>&nbsp;</p>
            <label class="alerta" id="indicador2"></label>
        </form>
    </div>
</div>
<div class="banner">
    <h2>Luz <br />Ambiental</h2>
    <p>&nbsp;</p>
    <div class="image_wrapper">
        <p></p>
        <p>&nbsp;</p>
        <form id="form1" name="form1" method="post" action="">
            <p>
                <label for="luzAmbiental"></label>
                <center><label class="textbox" id="luzAmbiental"></label></center>
            </p>
            <p>&nbsp;</p>
            <label class="alerta" id="indicador3"></label>
        </form>
    </div>
</div>
<div class="banner banner_last">
    <h2>Humedad <br />Suelo</h2>
    <p>&nbsp;</p>
    <div class="image_wrapper">
        <p></p>
        <p>&nbsp;</p>
        <form id="form4" name="form4" method="post" action="">
            <p>
                <label for="humedadSuelo"></label>
                <center><label class="textbox" id="humedadSuelo"></label></center>
            </p>
            <p>&nbsp;</p>
            <label class="alerta" id="indicador4"></label>
        </form>
    </div>
</div>

<p>&nbsp;</p>

```

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<hr />
</div>
</div>

<div id="footer_wrapper">
<div id="footer">
<div class="logo">Disẽado por astudillo y cadena</div>
<ul>
<li>inicio</li>
<li>recomendaciones</li>
<li>tips</li>
</ul>
</div>
</div></body>
</html>
```

Recomendaciones.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="UTF-8" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="sweetalert/dist/sweetalert.css">
<script src="sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript" src="alertaIndicadores.js"></script>
</head>

<body>
<div id="container">
<div id="containertop"></div>
<h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
<hr />
<div id="navigation_wrapper">
<ul id="navigation">
<li><a href="inicio.php" title="Inicio">Inicio</a></li>
<li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
<li><a href="tips.php" title="Tips">Tips</a></li>
</ul>
</div>

<div id="banner_wrapper">
<div class="banner">
<h2>Temperatura <br /> Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
```



```
<form id="form2" name="form2" method="post" action="">
  <p>
    <label for="temperaturaAmbiental"></label>
    <center><label class="textbox" id="temperaturaAmbiental"></label></center>
  </p>
  <p>&nbsp;</p>
  <label class="alerta" id="indicador1"></label>
</form>
</div>
<div class="banner">
  <h2>Humedad <br />Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form3" name="form3" method="post" action="">
      <p>
        <label for="humedadAmbiental"></label>
        <center><label class="textbox" id="humedadAmbiental"></label></center>
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador2"></label>
    </form>
  </div>
</div>
<div class="banner">
  <h2>Luz <br />Ambiental</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form1" name="form1" method="post" action="">
      <p>
        <label for="luzAmbiental"></label>
        <center><label class="textbox" id="luzAmbiental"></label></center>
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador3"></label>
    </form>
  </div>
</div>
<div class="banner banner_last">
  <h2>Humedad <br /> Suelo</h2>
  <p>&nbsp;</p>
  <div class="image_wrapper">
    <p></p>
    <p>&nbsp;</p>
    <form id="form4" name="form4" method="post" action="">
      <p>
        <label for="humedadSuelo"></label>
        <center><label class="textbox" id="humedadSuelo"></label></center>
      </p>
      <p>&nbsp;</p>
      <label class="alerta" id="indicador4"></label>
    </form>
  </div>
</div>
```

```

    </form>
  </div>
</div>

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<hr />
</div>

<div id="services">
  <h2>Te recomendamos...<span></span></h2>
  <div id="servicescontent">
    <form      class="form"      id="form1"      name="form1"      method="post"
    action="recomendacionFinal.php">
      <p>
        <label class="textPregunta" >¿Espacio disponible para su jardín?</label>
      </p>
      <p>&nbsp;</p>
      <p>
        <label class="radioPregunta">
          <input      type="radio"      name="radioPreguntaUno"      value="árbol"
id="radioPreguntaUno_0" />
          Amplio </label>

          <label class="radioPregunta">
            <input      type="radio"      name="radioPreguntaUno"      value="arbusto"
id="radioPreguntaDos_0" />
            Moderado </label>

            <label class="radioPregunta">
              <input      type="radio"      name="radioPreguntaUno"      value="herbácea"
id="radioPreguntaTres_1" />
              Limitado</label>
            </p>
            <p>&nbsp;</p>
            <p>
              <label class="textPregunta" >¿El jardín va a ser cubierto?</label>
            </p>
            <p>&nbsp;</p>
            <p>
              <label class="radioPregunta">
                <input      type="radio"      name="radioPreguntaDos"      value="moderada"
id="radioPreguntaCuatro_0" />
                Si </label>
              <label class="radioPregunta">
                <input      type="radio"      name="radioPreguntaDos"      value="abundante"
id="radioPreguntaCinco_1" />
                No</label>
              </p>
              <p>&nbsp;</p>

              <p>
                <label class="textPregunta" >¿Disponibilidad de tiempo para el cuidado de la
planta?</label>

```

```

</p>
<p>&nbsp;</p>
<p>
  <label class="radioPregunta">
    <input type="radio" name="radioPreguntaTres" value="3" id="radioPreguntaSeis_0" />
    Amplio </label>
  <label class="radioPregunta">
    <input type="radio" name="radioPreguntaTres" value="2" id="radioPreguntaSiete_1" />
    Moderado</label>
  <label class="radioPregunta">
    <input type="radio" name="radioPreguntaTres" value="1" id="radioPreguntaOcho_1" />
    Limitado</label>
</p>
<p>&nbsp;</p>
<p>
  <label class="textPregunta" >¿ Vive en valles?</label>
</p>
<p>&nbsp;</p>
<p>
  <label class="radioPregunta">
    <input type="radio" name="radioPreguntaCuatro" value="19"
id="radioPreguntaNueve_0" />
    Si </label>
  <label class="radioPregunta">
    <input type="radio" name="radioPreguntaCuatro" value="18"
id="radioPreguntaDiez_1" />
    No</label>
</p>

  <p>
    <input type="submit" class="button" name="botonSubmitRecomendaciones"
id="botonSubmitRecomendaciones" value="Recomendar" />
  </p>
  <p> <br />
</p>
</form>
</div>
<span class="bottom_rounded"></span>
<hr />
</div>
</div>

<div id="footer_wrapper">
  <div id="footer">
    <div class="logo">Dise&ntilde;ado por astudillo y cadena</div>
    <ul>
      <li>inicio</li>
      <li>recomendaciones</li>
      <li>tips</li>
    </ul>
  </div>
</div></body>
</html>

```

RecomendacionFinal.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="UTF-8" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="sweetalert/dist/sweetalert.css">
<script src="sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript" src="alertaIndicadores.js"></script>
</head>

<body>
<div id="container">
<div id="containertop"></div>
<h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
<hr />
<div id="navigation_wrapper">
<ul id="navigation">
<li><a href="inicio.php" title="Inicio">Inicio</a></li>
<li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
<li><a href="tips.php" title="Tips">Tips</a></li>
</ul>
</div>

<div id="banner_wrapper">
<div class="banner">
<h2>Temperatura <br /> Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form2" name="form2" method="post" action="">
<p>
<label for="temperaturaAmbiental"></label>
<center><label class="textbox" id="temperaturaAmbiental"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador1"></label>
</form>
</div>
</div>
<div class="banner">
<h2>Humedad <br />Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form3" name="form3" method="post" action="">
<p>
<label for="humedadAmbiental"></label>
```

```
<center><label class="textbox" id="humedadAmbiental"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador2"></label>
</form>
</div>
<div>
<div class="banner">
<h2>Luz <br />Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form1" name="form1" method="post" action="">
<p>
<label for="luzAmbiental"></label>
<center><label class="textbox" id="luzAmbiental"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador3"></label>
</form>
</div>
</div>
<div class="banner banner_last">
<h2>Humedad <br /> Suelo</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form4" name="form4" method="post" action="">
<p>
<label for="humedadSuelo"></label>
<center><label class="textbox" id="humedadSuelo"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador4"></label>
</form>
</div>
</div>

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<hr />
</div>
```

```
<div id="services">
<h2>Te recomendamos...<span></span></h2>
<div id="servicescontent">
<?php
include "conexion.php";
$pregunta1= $_POST['radioPreguntaUno'];
$pregunta2= $_POST['radioPreguntaDos'];
$pregunta3= $_POST['radioPreguntaTres'];
$pregunta4= $_POST['radioPreguntaCuatro'];
```

```

        if($pregunta4==19)
        {
            $queryRecomendacion = "select NOMBREPLANTA, INFOEXTRAPLANTA
FROM PLANTA WHERE TIPOPLANTA = ".$pregunta1." AND LUZPLANTA =
".$pregunta2." AND RIEGOSEMANALPLANTA = ".$pregunta3." AND
TEMPERATURAMAXPLANTA >= ".$pregunta4."" ;
            $resultadoRecomendacion = mysqli_query($conexion,$queryRecomendacion) or
die(mysqli_error($conexion));
        }
        else
        {
            $queryRecomendacion = "select NOMBREPLANTA,
INFOEXTRAPLANTA FROM PLANTA WHERE TIPOPLANTA = ".$pregunta1." AND
LUZPLANTA = ".$pregunta2." AND RIEGOSEMANALPLANTA = ".$pregunta3." AND
TEMPERATURAMAXPLANTA <= ".$pregunta4."" ;
            $resultadoRecomendacion = mysqli_query($conexion,$queryRecomendacion) or
die(mysqli_error($conexion));
        }
    ?>
    <center><Table BORDER="1" cellspacing="3">
    <TR>
    <TH>NOMBRE PLANTA</TH>
    <TH>INFORMACIÓN ADICIONAL</TH>
    </TR>
    <?php
    while($auxPlantas = mysqli_fetch_array($resultadoRecomendacion))
    { ?>
    <TR>
    <TD><?PHP echo $auxPlantas['NOMBREPLANTA'];?></TD>
    <TD><?PHP echo $auxPlantas['INFOEXTRAPLANTA'];?></TD>
    </TR>
    <?php } ?>
    </Table></center>
    </div>
    <span class="bottom_rounded"></span>
    <hr />
    </div>
    </div>

    <div id="footer_wrapper">
    <div id="footer">
    <div class="logo">Disenado por astudillo y cadena</div>
    <ul>
    <li>inicio</li>
    <li>recomendaciones</li>
    <li>tips</li>
    </ul>
    </div>
    </div></body>
    </html>

```

Tips.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="UTF-8" />
<title>Jardines Urbanos</title>
<link href="stylesheets/common.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="sweetalert/dist/sweetalert.css">
<script src="sweetalert/dist/sweetalert.min.js"></script>
<script type="text/javascript" src="alertaIndicadores.js"></script>
</head>
<body>
<div id="container">
<div id="containertop"></div>
<h1><a href="inicio.php" title="Jardines Urbanos">JARDINES URBANOS</a></h1>
<hr />
<div id="navigation_wrapper">
<ul id="navigation">
<li><a href="inicio.php" title="Inicio">Inicio</a></li>
<li><a href="recomendaciones.php" title="Recomendaciones">Recomendaciones</a></li>
<li><a href="tips.php" title="Tips">Tips</a></li>
</ul>
</div>

<div id="banner_wrapper">
<div class="banner">
<h2>Temperatura <br /> Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form2" name="form2" method="post" action="">
<p>
<label for="temperaturaAmbiental"></label>
<center><label class="textbox" id="temperaturaAmbiental"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador1"></label>
</form>
</div>
</div>
<div class="banner">
<h2>Humedad <br />Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form3" name="form3" method="post" action="">
<p>
<label for="humedadAmbiental"></label>
<center><label class="textbox" id="humedadAmbiental"></label></center>
</p>
</form>
</div>
</div>
```

```
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador2"></label>
</form>
</div>
</div>
<div class="banner">
<h2>Luz <br /> Ambiental</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form1" name="form1" method="post" action="">
<p>
<label for="luzAmbiental"></label>
<center><label class="textbox" id="luzAmbiental"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador3"></label>
</form>
</div>
</div>
<div class="banner banner_last">
<h2>Humedad <br /> Suelo</h2>
<p>&nbsp;</p>
<div class="image_wrapper">
<p></p>
<p>&nbsp;</p>
<form id="form4" name="form4" method="post" action="">
<p>
<label for="humedadSuelo"></label>
<center><label class="textbox" id="humedadSuelo"></label></center>
</p>
<p>&nbsp;</p>
<label class="alerta" id="indicador4"></label>
</form>
</div>
</div>

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p></p>
<p>&nbsp;</p>
```



```
<hr />
</div>
</div>

<div id="footer_wrapper">
  <div id="footer">
    <div class="logo">Diseno por astudillo y cadena</div>
    <ul>
      <li>inicio</li>
      <li>recomendaciones</li>
      <li>tips</li>
    </ul>
  </div>
</div></body>
</html>
```

Humedad.php

```
<?php
echo file_get_contents('http://192.168.100.15/arduino/humedad/13', NULL, NULL, 0, 2);
?>
```

HumedadSuelo.php

```
<?php
echo file_get_contents('http://192.168.100.15/arduino/suelo/0', NULL, NULL, 0, 10);
?>
```

Luz.php

```
<?php
echo file_get_contents('http://192.168.100.15/arduino/luz/3', NULL, NULL, 0, 10);
?>
```

Temperatura.php

```
<?php
echo file_get_contents('http://192.168.100.15/arduino/temperatura/13', NULL, NULL, 0, 2);
?>
```

ANEXO 5: Manual de usuario

Requisitos de Software

Contar con:

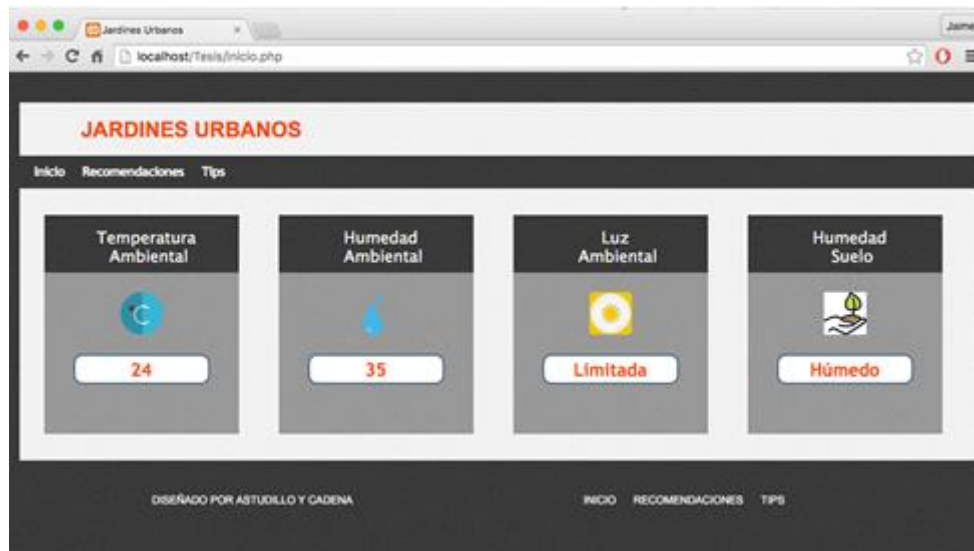
- Sistema operativo Windows, Mac OSX, Linux
- Navegador actualizado. (Chrome, Mozilla Firefox, Safari)
- Servidor Web Apache (XAMP, WAMP)
- MySQL (XAMPP, WAMP)

Ingreso al sistema

Para poder acceder al sistema para la administración de jardines urbanos, es necesario ingresar al siguiente URL:

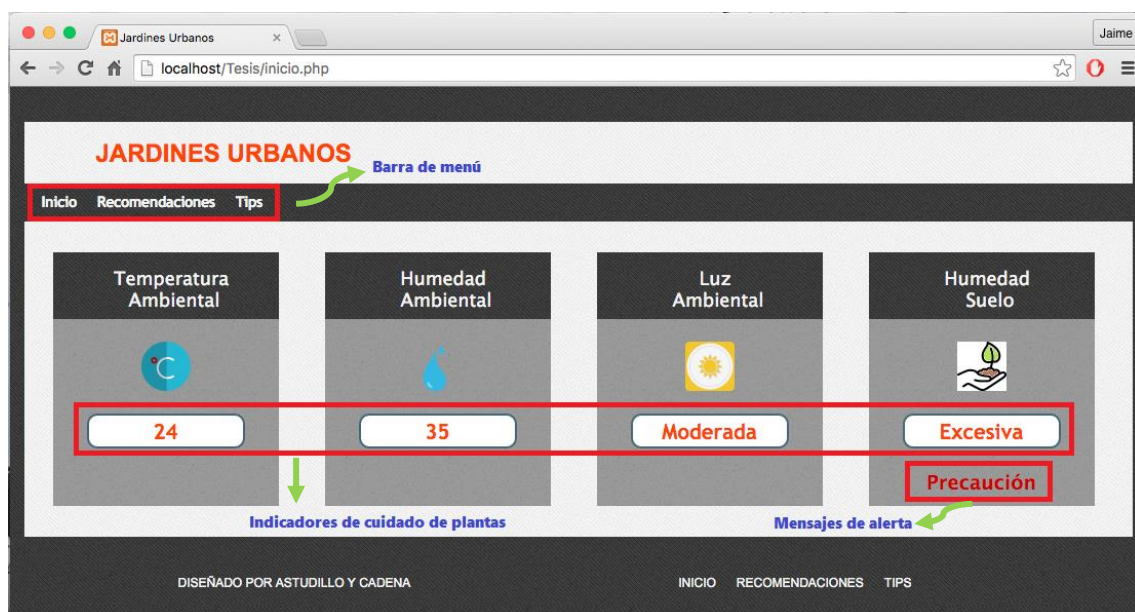
<http://<Dirección de la pagina>/Tesis/inicio.php>

<http://localhost/Tesis/inicio.php>



Funcionalidades Generales

El sistema cuenta con una barra de menú de opciones con la que se podrá navegar en las diferentes páginas. Además cuenta con un área destinada para presentar los indicadores de cuidado de plantas, la misma que se actualizará cada cierto tiempo. Cada indicador cuenta con una zona designada para mostrar un mensaje de alerta de acuerdo a sus límites.



Indicadores de cuidado de plantas



Como se puede apreciar en la imagen, los indicadores que se presentan son: temperatura ambiental, humedad ambiental, luz ambiental y humedad de suelo. Cada uno muestra información procesada, recibida de los sensores respectivos.

En caso de existir la necesidad de revisar el estado de las plantas, se mostrará un mensaje indicando cuál de los indicadores requiere atención.

Recomendaciones

The image is a screenshot of a web browser displaying a form titled 'Te recomendamos...' (We recommend...). The form is set against a dark background with a light grey central panel. It contains five questions, each with radio button options: 1. '¿Espacio disponible para su jardín?' (Available space for your garden?) with options 'Amplio' (Wide), 'Moderado' (Moderate), and 'Limitado' (Limited). 2. '¿El jardín va a ser cubierto?' (Will the garden be covered?) with options 'Si' (Yes) and 'No'. 3. '¿Disponibilidad de tiempo para el cuidado de la planta?' (Availability of time for plant care?) with options 'Amplio', 'Moderado', and 'Limitado'. 4. '¿Vive en valles?' (Do you live in valleys?) with options 'Si' and 'No'. A blue 'RECOMENDAR' (Recommend) button is located at the bottom right of the form. Below the form, a footer contains the text 'DISEÑADO POR ASTUDILLO Y CADENA' and navigation links 'INICIO', 'RECOMENDACIONES', and 'TIPS'.

Mediante la barra de menú, dirigirse a la opción de Recomendaciones. Se podrá visualizar un cuestionario que una vez completado, se podrá presionar el botón Recomendar. De acuerdo a las respuestas dadas, se presentarán las sugerencias de plantas.

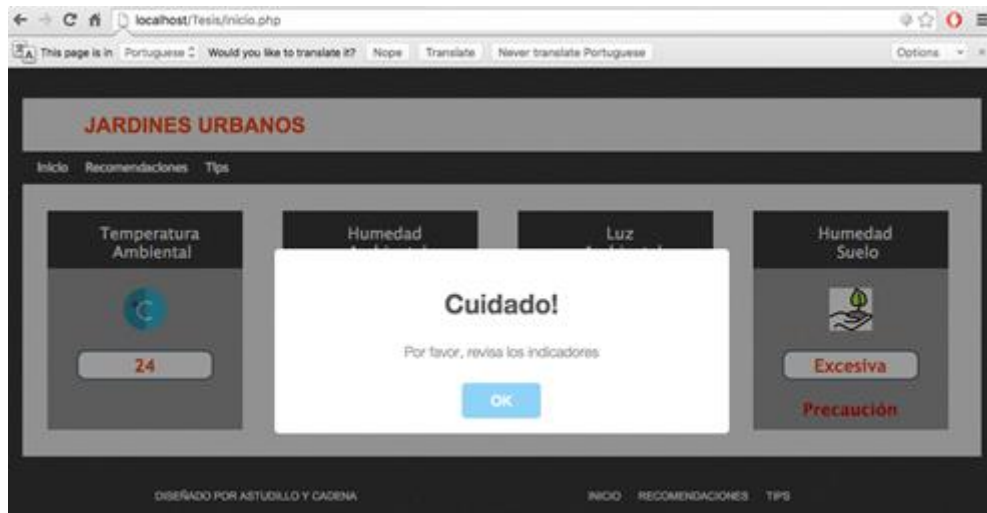


Tips



Mediante la barra de menú, dirigirse a la opción de Tips. Se podrá visualizar tips que sirvan de guía para un mejor cuidado de los jardines.

Alertas



Las alertas se mostrarán una vez que los indicadores de plantas estén fuera del rango recomendado. Además se mostrará un mensaje de alerta en la zona designada de cada indicador.